Final Report

Knowledge Discovery from Growing Social Networks

AFOSR/AOARD Reference Number: AOARD-08-4027

AFOSR/AOARD Program Manager: **Hiroshi Motoda**, Ph.D.

Period of Performance: 13 Dec. 07 – 12 Dec. 09

Submission Date: 19 Dec. 09

PI: Kazumi Saito

University of Shizuoka, 52-1 Yada, Suruga-ku, Shizuoka 422-8526 Japan

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
24 DEC 2009	**FInal**	**13-12-2007 to 12-12-2009**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Knowledge Discovery from Growing Social Networks	**FA48690814027**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
Kazumi Saito	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)	8. PERFORMING ORGANIZATION REPORT NUMBER
School of Administration and Informatics, University of Shizuoka,52-1 Yada, Suruga-ku,Shizuoka 422-8526,Shizuoka ,JP,422-8526	**N/A**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
AOARD, UNIT 45002, APO, AP, 96337-5002	**AOARD**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)
	AOARD-084027

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The project explored mathematical models to explain, control and visualize a wide variety of information diffusion processes. The main results are the following six. 1) A very efficient method for minimizing the propagation of undesirable things by blocking a limited number of links in a network. 2) An effective visualization method for understanding a complex network, in particular its dynamical aspect such as information diffusion process. 3) A new scheme for empirical study to explore the behavioral characteristics of representative information diffusion models. 4) An effective method for ranking influential nodes in complex social networks by estimating diffusion probabilities from observed information diffusion data using the popular independent cascade (IC) model. 5) A very efficient method for discovering the influential nodes in a social network under the susceptible/infected/susceptible (SIS) model. 6) A new method for learning continuous-time information diffusion model for social behavioral data analysis.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**154**	

(1) Objectives: Most complex networks, such as the World Wide Web, grow over time, and such growth is usually characterized by highly distributed phenomena. However, the complexity and distributed nature of those networks does not imply that its growth is chaotic or unpredictable. Just as natural scientists discover laws and create models for their fields, so one can , in principle, find empirical regularities and develop explanatory accounts of changes in the network. In the case of the World Wide Web, such predictive knowledge would be valuable for anticipating computing needs, social trends, and market opportunities.

We can now obtain digital traces of human social interaction with time stamp information in a wide variety of on-line settings, such as Blog (Weblog) communications, email exchanges, etc.. Such social interaction can be naturally represented as a large-scale social network that grows over time, where nodes (vertices) correspond to people or some social entities, and links (edges) correspond to social interaction between them. Clearly these growing social networks reflect complex social structures and distributed social trends. Thus, it seems worth an effort to attempt to find empirical regularities and develop explanatory accounts of changes in the social networks. Namely, such attempts would be valuable for understanding social structures and trends, and inspiring us the discovery of new knowledge and insights into underlying social interaction. We extensively carry out research on computational methods for the discovery of knowledge from growing social networks.

(2) Status of effort: We have uncovered that probabilistic models of information diffusion processes over social networks play an essential role for the discovery of knowledge. Thus, we carried out research on mathematical models for enabling us to explain, control and visualize wider variety of information diffusion processes. Especially, it is highly expected that this kind of mathematical studies using large-scale networks such as a blog communication network can bridge a gap between empirical social networks analyses and fundamental mathematics. In the first year, we derived a very efficient method for minimizing the propagation of undesirable things by blocking a limited number of links in a network. In addition, we developed an effective visualization method for understanding a complex network, in particular its dynamical aspect such as information diffusion process. Furthermore, we proposed a new scheme for empirical study to explore the behavioral characteristics of representative information diffusion models. In the second year, we developed an effective method for ranking influential nodes in complex social networks by estimating diffusion probabilities from observed information diffusion data using the popular independent cascade (IC) model. In addition, we derived a very efficient method for discovering the influential nodes in a social network under the *susceptible/infected/susceptible (SIS) model*. Furthermore, we proposed a new method for learning continuous-time information diffusion model for social behavioral data analysis.

(3) Abstract: First, we addressed the problem of minimizing the propagation of undesirable things, such as computer viruses or malicious rumors, by blocking a limited number of links in a network, a converse problem to the influence maximization problem of finding the most influential nodes in a social network for information diffusion. This minimization problem is another approach to the problem of preventing the spread of

information. We derived a method for efficiently finding a good approximate solution to this problem based on a naturally greedy strategy. Using large real networks, we demonstrated experimentally that the proposed method significantly outperforms conventional link-removal methods. We also showed that unlike the strategy of removing nodes, blocking links between nodes with high out-degrees is not necessarily effective.

Second, we addressed the problem of effective visualization for understanding a complex network, in particular its dynamical aspect such as information diffusion process. Existing node embedding methods are all based solely on the network topology and sometimes produce counter-intuitive visualization. We developed a new node embedding method based on conditional probability that explicitly addresses diffusion process using either the IC (Independent Cascade) or LT (Linear Threshold) models as a cross-entropy minimization problem, together with two label assignment strategies that can be simultaneously adopted. Numerical experiments were performed on two large real networks, one represented by a directed graph and the other by an undirected graph. The results clearly demonstrated the advantage of the developed method over conventional spring model and topology-based cross-entropy methods, especially for the case of directed networks.

Third, we attempted to answer a question "What does information diffusion model tell about social network structure?" To this end, we proposed a new scheme for empirical study to explore the behavioral characteristics of representative information diffusion models such as the IC model and the LT model on large networks with different community structure. To change community structure, we first construct a GR (Generalized Random) network from an originally observed network by randomly rewiring links of the original network without changing the degree of each node. Then we plot the expected number of influenced nodes based on an information diffusion model with respect to the degree of each information source node. Using large real networks, we empirically found that our proposal scheme uncovered a number of new insights. Most importantly, we showed that community structure more strongly affects information diffusion processes of the IC model than those of the LT model. Moreover, by visualizing these networks, we gave some evidence that our claims are reasonable.

Forth, we addressed the problem of ranking influential nodes in complex social networks by estimating diffusion probabilities from observed information diffusion data using the IC model. For this purpose we formulated the likelihood for information diffusion data which is a set of time sequence data of active nodes and propose an iterative method to search for the probabilities that maximizes this likelihood. We apply this to two real world social networks in the simplest setting where the probability is uniform for all the links, and show that the accuracy of the probability estimation is outstandingly good, and further show that the proposed method can predict the high ranked influential nodes much more accurately than the well studied conventional four heuristic methods.

Fifth, we addressed the problem of efficiently discovering the influential nodes in a social network under the SIS model, a diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property. We solved this problem by constructing a layered graph from the original social network with each layer added on top as the time proceeds, and applying the bond percolation with pruning and burnout strategies. We experimentally demonstrated that the proposed method gives much better solutions than the conventional

methods that are solely based on the notion of centrality for social network analysis using two large-scale real-world networks (a blog network and a wikipedia network). We further showed that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis and confirm this by experimentation. The properties of the influential nodes discovered were substantially different from those identified by the centrality-based heuristic methods.

Finally, we addressed the problem of estimating the parameters for a continuous time delay independent cascade (CTIC) model, a more realistic model for information diffusion in complex social network, from the observed information diffusion data. For this purpose we formulated the rigorous likelihood to obtain the observed data and propose an iterative method to obtain the parameters (time-delay and diffusion) by maximizing this likelihood. We applied this method first to the problem of ranking influential nodes using the network structure taken from two real world web datasets and showed that the proposed method can predict the high ranked influential nodes much more accurately than the well studied conventional four heuristic methods, and second to the problem of evaluating how different topics propagate in different ways using a real world blog data and showed that there are indeed differences in the propagation speed among different topics.

(4) Personnel Supported:

Masahiro Kimura / Department of Electronics and Informatics, Ryukoku University

(5) Publications:

1. Masahiro Kimura, Kazumasa Yamakawa, Kazumi Saito and Hiroshi Motoda, Community Analysis of Influential Nodes for Information Diffusion on a Social Network, Proc. of the International Joint Conference on Neural Networks (WCCI2008), pp.1359--1364, 2008.
2. Masahiro Kimura, Kazumi Saito and Hiroshi Motoda, Minimizing the Spread of Contamination by Blocking Links in a Network, Proc. of the Twenty-Third Conference on Artificial Intelligence (AAAI2008), pp.1175--1180, 2008.
3. Kazumi Saito, Masahiro Kimura and Hiroshi Motoda, Effective Visualization of Information Diffusion Process over Complex Networks, Proc. of the Eighteenth European Conference on Machine Learning (ECML2008), LNAI 5212 , pp.326--341, 2008.
4. Masahiro Kimura, Kazumi Saito and Hiroshi Motoda, Solving the Contamination Minimization Problem on Networks for the Linear Threshold Model, Proc. of the 10th Pacific Rim International Conference on Artificial Intelligence (PRICAI2008), pp.977--984, 2008.
5. Takayasu Fushimi, Takashi Kawazoe, Kazumi Saito, Masahiro Kimura and Hiroshi Motoda, What Does an Information Diffusion Model Tell about Social Network Structure?, Proc. of the 2008 Pacific Rim Knowledge Acquisition Workshop (PKAW2008), pp.288--299, 2008.
6. Masahiro Kimura, Kazumi Saito and Hiroshi Motoda, Blocking Links to Minimize Contamination Spread in a Social Network, ACM Transactions on Knowledge Discovery from Data, Vol.3, No.2, Article 9, pp.1--23, 2009.
7. Masahiro Kimura, Kazumi Saito and Hiroshi Motoda, Finding Influential Nodes in a Social Network from Information Diffusion Data, Proc. of the Second International

Workshop on Social Computing, Behavioral Modeling, and Prediction (SBP2009), pp.138--145, 2009.
8. Masahiro Kimura, Kazumi Saito and Hiroshi Motoda, Efficient Estimation of Influence Functions for SIS Model on Social Networks, Proc. of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI2009), , pp.2046--2051, 2009.
9. Kazumi Saito, Masahiro Kimura and Hiroshi Motoda, Discovering Influential Nodes for SIS models in Social Networks, Proc. of the 12th International Conference on Discovery Science (DS2009), pp.302--316, 2009.
10. Kazumi Saito, Masahiro Kimura, Kouzou Ohara and Hiroshi Motoda, Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis, Proc. of the First Asian Conference on Machine Learning (ACML2009), pp.322--337, 2009.
11. Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda, Extracting Influential Nodes on a Social Network for Information Diffusion, Data Mining and Knowledge Discovery, (in press).

(6) Interactions:

1. Research meeting with M. Kimura and H. Motoda in Shizuoka (20 Dec 2007).
2. Research meeting with M. Kimura and H. Motoda in Tokyo (15 Jan 2008).
3. Research meeting with M. Kimura and H. Motoda in Tokyo (28 Feb 2008).
4. Research meeting with M. Kimura and H. Motoda in Tokyo (18 Mar 2008).
5. Research meeting with M. Kimura and H. Motoda in Osaka (15 May 2008).
6. Research meeting with T. Lyons and H. Motoda in Tokyo (19 May 2008).
7. Research presentation by M. Kimura at WCCI2008 in Hong Kong (5 Jun 2008).
8. Research meeting with H. Motoda in Hokkaido (15 Jun 2008).
9. Research presentation by M. Kimura at AAAI2008 in Chicago (15 Jul 2008).
10. Research meeting with M. Kimura and H. Motoda in Tokyo (21Jul 2008).
11. Research meeting with M. Kimura and H. Motoda in Tokyo (22 Aug 2008).
12. Research presentation by K Saito at ECML2008 in Antwerp (18 Sep 2008).
13. Research meeting with P. Friedland and H. Motoda in Tokyo (11Nov 2008).
14. Research meeting with M. Kimura and H. Motoda in Tokyo (17 Nov 2008).
15. Research presentation by T. Fushimi at PKAW2008 in Hanoi (16 Dec 2008).
16. Research presentation by M. Kimura at PRICAI2008 in Hanoi (18 Dec 2008).
17. Research meeting with M. Kimura and H. Motoda in Tokyo (12 Jan 2009).
18. Research presentation by K Saito at AFOSR Program Review in Arlington (27 Jan 2009).
19. Research meeting with M. Kimura and H. Motoda in Tokyo (16 Mar 2009).
20. Research presentation by M. Kimura at SBP2009 in Phoenix (31 Mar 2008).
21. Research meeting with M. Kimura and H. Motoda in Tokyo (18 Apr 2009).
22. Research meeting with M. Kimura, K. Ohara and H. Motoda in Tokyo (16 May 2009).
23. Research meeting with M. Kimura, K. Ohara and H. Motoda in Kagawa (18 Jun 2009).
24. Research meeting with M. Kimura, K. Ohara and H. Motoda in Tokyo (4 Aug 2009).
25. Research meeting with M. Kimura, K. Ohara and H. Motoda in Tokyo (5 Sep 2009).
26. Research meeting with M. Kimura, K. Ohara and H. Motoda in Osaka (29 Sep 2009).
27. Research presentation by K Saito at DS2009 in Porto (4 Oct 2008).

28. Research presentation by K Saito at ACML2009 in Nanjin (3 Nov 2008).
29. Research meeting with M. Kimura, K. Ohara and H. Motoda in Kanagawa (9 Nov 2009).
30. Research meeting with T. Lyons and H. Motoda in Tokyo (17 Nov 2008).
31. Research meeting with M. Kimura, K. Ohara and H. Motoda in Tokyo (19 Dec 2009).

(7) New: None.

(8) Honors/Awards: None.

(9) Archival Documentation: The above eleven papers are attached.

(10) Software and/or Hardware (if they are specified in the contract as part of final deliverables): None.

Community Analysis of Influential Nodes for Information Diffusion on a Social Network

Masahiro Kimura, Kazumasa Yamakawa, Kazumi Saito, and Hiroshi Motoda

***Abstract*— We consider the problem of finding influential nodes for information diffusion on a social network under the independent cascade model. It is known that the greedy algorithm can give a good approximate solution for the problem. Aiming to obtain efficient methods for finding better approximate solutions, we explore what structual feature of the underlying network is relevant to the *greedy solution* that is the approximate solution by the greedy algorithm. We focus on the SR-community strucutre, and analyze the greedy solution in terms of the SR-community structure. Using real large social networks, we experimentally demonstrate that the SR-community structure can be more strongly correlated with the greedy solution than the community structure introduced by Newman and Leicht.**

I. Introduction

Recently, considerable attention has been devoted to social network analysis [9], [14], [1], [2], [8], [13], [7], since the rise of the Internet and the World Wide Web has enabled us to collect real large social networks. Here, a social network is the network of relationships and interactions among social entities such as individuals, organizations and groups. Examples include blog networks, collaboration networks, and email networks.

A social network plays an important role for the spread of information since a piece of information can propagate from one node to another node through a link on the network in the form of "word-of-mouth" communication [3]. Thus, it is an important research issue to find influential nodes for information diffusion on a social network in terms of sociology and "viral marketing". In fact, researchers [5], [6] have recently studied a combinatorial optimization problem called the *influence maximization problem* on a network under the *independent cascade (IC) model* that is a widely-used fundamental probabilistic model of information diffusion. Here, the influence maximization problem of size k is the problem of extracting a set of k nodes to target for initial activation such that it yields the largest expected spread of information, where k is a given positive integer. Kempe *et al.* [5] experimentally showed on large collaboration networks that the greedy algorithm can give a good approximate solution for the influence maximization problem under the IC model. We refer to the approximate solution obtained by the greedy algorithm as the *greedy solution*. Using an analysis framework based on submodular functions, Kempe *et al.* [5] mathematically proved a performance guarantee of the greedy solution. Moreover, Kimura *et al.* [6] presented a method of efficiently estimating the greedy solution on the basis of bond percolation and graph theory. However, it is desirable to construct efficient methods of obtaining better approximate solutions for the influence maximization problem on a network under the IC model. Towards this aim, it is important to understand what structural feature of the underlying network is correlated with the greedy solution.

As a structual feature of a given network, we focus on the *SR-community structure* $\mathcal{U} = \langle U_m;\, m = 1, 2, 3, \cdots \rangle$ [15] that is a sequence of densely connected sets of nodes in the network. Here, the mth SR-community U_m is defined as the set of nodes in the network that maximizes the average number of links within it after removing all the links within U_j, $(j = 0, \cdots, m-1)$, where U_0 is the empty set $\emptyset$. In this paper, we analyze the greedy solution for the influence maximization problem under the IC model in terms of the SR-community structure $\mathcal{U}$. For the influence maximization problem of size k, we extract the minimal sequence of SR-communities in $\mathcal{U}$, $\mathcal{U}_k = \langle U_m; m = 1, \cdots, M_k \rangle$, such that it covers the greedy solution, and investigate the similarity between the set of nodes influenced by each node v_i in the greedy solution and the SR-community in $\mathcal{U}_k$ that corresponds to the node v_i. On the basis of this manner, we quantify the strength of the correlation between the greedy solution and the SR-community struture. Using real large social networks, we experimentally demonstrate that unlike the community structure introduced by Newman and Leicht [12], the SR-community structure can be strongly correlated wirh the greedy solution.

II. Influential Nodes for Information Diffusion

Throughout this paper, we consider a social network represented by an undirected graph, and discuss the spread of a certain information through the network under the IC model by regarding those undirected links as bidirectional ones. We call nodes *active* if they have accepted the information.

A. Independent Cascade Model

We define the IC model. In this model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is

Masahiro Kimura is with the Department of Electronics and Informatics, Faculty of Science and Technology, Ryukoku University, Otsu 520-2194, Japan (phone: +81 77 543 7406; fax: +81 77 543 7749; email: kimura@rins.ryukoku.ac.jp).

Kazumasa Yamakawa is with the Division of Electronics and Informatics, Graduate School of Science and Technology, Ryukoku University, Otsu 520-2194, Japan (email: t07m025@mail.ryukoku.ac.jp).

Kazumi Saito is with the School of Administration and Informatics, University of Shizuoka, Shizuoka 422-8526, Japan (email: k-saito@u-shizuoka-ken.ac.jp).

Hiroshi Motoda is with the Institute of Scientific and Industrial Research, Osaka University, Osaka 567-0047, Japan (email: motoda@ar.sanken.osaka-u.ac.jp).

assumed that nodes can switch from being inactive to being active, but cannot switch from being active to being inactive. Given an initial set X of active nodes, we assume that the nodes in X have first become active at step 0, and all the other nodes are inactive at step 0. We specify a real value $\beta_{u,v} \in [0,1]$ for each directed link (u,v) in advance. Here, $\beta_{u,v}$ is referred to as the *propagation probability* through link (u,v).

When an initial set X of active nodes is given, the diffusion process proceeds in the following way. When node u first becomes active at step t, it is given a single chance to activate each currently inactive neighbor v, and succeeds with probability $\beta_{u,v}$. If u succeeds, then v will become active at step $t+1$. If multiple parents of v first become active at step t, then their activation attempts are sequenced in an arbitrary order, but performed at step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set X, let $\sigma(X)$ denote the expected number of active nodes at the end of the random process in the IC model. We call $\sigma(X)$ the *influence degree* of initial active set X.

B. Influence Maximization Problem

We consider the influence maximization problem of size k under the IC mode. Let S be the set of all the nodes in the network. The problem is defined as follows: Given a positive integer k, find a set X_k^* of k nodes to target for initial activation such that $\sigma(X_k^*) \geq \sigma(Y)$ for any set Y of k nodes. To approximately solve this optimization problem, we consider the following greedy algorithm:

1) Set $X \leftarrow \emptyset$.
2) **for** $i = 1$ to k **do**
3) Choose a node $v_i \in V$ maximizing $\sigma(X \cup \{v\})$, $(v \in S \setminus X)$.
4) Set $X \leftarrow X \cup \{v_i\}$.
5) **end for**

Let S_k denote the set of k nodes obtained by this algorithm. We call S_k the *greedy solution* of the influence maximization problem of size k.

Using large collaboration networks, Kempe *et al.* [5] experimetally demonstrated that the greedy solution S_k outperforms the approximate solutios obtained by the high-degree and centrality heuristics that are commonly used in the sociology literature. It is also known that

$$\sigma(S_k) \geq \left(1 - \frac{1}{e}\right) \sigma(X_k^*),$$

that is, a performance guarantee of the greedy solution S_k is obtained [5]. For any initial active set X, a good estimate of $\sigma(X)$ was conventionally obtained by simulating the random process of the IC model many times. Thus, any straightforward method to estimate the greedy solution S_k needed a large amount of computation on a large network. However, Kimura *et al.* [6] gave an efficient method for estimating S_k on the basis of bond percolation and graph theory. In this paper, using their method, we estimate the greedy solution S_k.

III. SR-community Structure

In this section, we define the SR-community structure, and describe a method for efficiently estimating it according to the work of Saito *et al.* [15].

A. Definition

Let $\mathbf{A}$ be the adjacency matrix of a network, and let

$$S = \{1, \cdots, N\}$$

be the set of all the nodes in the network. Namely, each (i,j)-element of the adjacency matrix, denoted by $A(i,j)$, is set to 1 if there exists a link (edge) between nodes i and j; otherwise 0. In this paper, we focus on undirected graphs without self-connections, i.e., $A(i,j) = A(j,i)$, $A(i,i) = 0$, $(i,j = 1, \cdots, N)$. For any subset of nodes, $T \subset S$, we can define *the average number of links within* T as follows:

$$G(T) = \frac{1}{2} \sum_{i \in T} \sum_{j \in T} \frac{A(i,j)}{|T|}, \tag{1}$$

where $|T|$ stands for the number of elements in T. First, let U_1 denote the subset of S that maximizes the average number of links within it (see, (1)). Next, for the network constructed through removing all the links within U_1 from the original network, let U_2 denote the subset of S that maximizes the average number of links within it (see, (1)). Next, for the network constructed thorugh removing all the links within U_1 and U_2 from the original network, let U_3 denote the subset of S that maximizes the average number of links within it (see, (1)). By repeatedly performing these procedures, we define the sequence of subsets of S,

$$\mathcal{U} = \langle U_m;\, m = 1, 2, 3, \cdots \rangle.$$

Here, $\mathcal{U}$ is called the *SR-community structure* of the original network, and each U_m is referred to as the *mth SR-community*. Note that the SR-community structure $\mathcal{U}$ represents a structual feature of the network.

In the case of a large network, any straightforward method for detecting the SR-community structure is likely to suffer from combinatorial explosion. To cope with such a large network, we employ the method presented by Saito *et al.* [15].

B. Relaxation problem

For a subset T of S , we define an N dimensional indicator vector $\mathbf{q}$ by setting $q(i) = 1$ if $i \in T$; otherwise $q(i) = 0$. Then we can rewrite (1) as follows:

$$G(\mathbf{q}) = \frac{1}{2} \frac{\mathbf{q}^T \mathbf{A} \mathbf{q}}{\mathbf{q}^T \mathbf{q}}, \tag{2}$$

where $\mathbf{q}^T$ stands for a transposed vector of $\mathbf{q}$. Now we consider a relaxation problem by letting $\mathbf{q}$ take continuous values. Then, according to the Rayleigh-Ritz theorem [4],

the solution of maximizing $G(\mathbf{q})$ is given by the principal eigenvector $\mathbf{q}^*$ of the adjacency matrix $\mathbf{A}$.

In order to obtain the eigenvector $\mathbf{q}*$, we employ the following procedure based on the power iteration [4].

E1. Initialize $\mathbf{q}^{(0)} = (1, \cdots, 1)^T$, and set $\tau \leftarrow 1$;
E2. Calculate $\widetilde{\mathbf{q}} = \mathbf{A}\mathbf{q}^{(\tau-1)}$ and $\mathbf{q}^{(t)} = \widetilde{\mathbf{q}}/\max_i \widetilde{q}_i$;
E3. Terminate if $\max_i |q^{(\tau)}(i) - q^{(\tau-1)}(i)| < \varepsilon$;
E4. Set $\tau \leftarrow \tau + 1$, and return to **E2.**.

Here a small positive parameter ε controls the termination condition, and we can obtain the final solution as $\mathbf{q}^* = \mathbf{q}^{(\tau)}$ after its termination. Since all the elements of $\mathbf{A}$ and $\mathbf{q}^{(0)}$ have non-negative values, we can guarantee that all the elements of $\widetilde{\mathbf{q}}$ also have non-negative values after any number of iterations. Moreover, due to the scaling operation in **E2**, we can guarantee that $0 \leq q^{(\tau)}(i) \leq 1$ for any τ and i. Thus we consider that the above formulation gives one of desirable relaxation solutions to the original problem.

C. Quantization problem

By ranking nodes according to the values of eigenvector elements, we can obtain a list of nodes, $R = [r(1), \cdots, r(N)]$, where $r(i)$ stands for a mapping from ranks to nodes. Note that $q^*(r(i)) \geq q^*(r(i+1))$ for any i. By considering a set of the top h nodes,

$$T(h) = \{r(i) : i = 1, \cdots, h\}, \tag{3}$$

we can calculate the average number of links within $T(h)$ as follows:

$$G(h) = \sum_{i=1}^{h-1} \sum_{j=i+1}^{h} \frac{A(r(i), r(j))}{h}. \tag{4}$$

In our method, instead of directly solving (1), we compute a node set $T(h^*)$, where h^* maximizes (4).

In order to efficiently calculate h^*, we utilize the following update formula:

$$G(h+1) = G(h) + \frac{\Delta(h+1) - G(h)}{h+1}, \tag{5}$$

where $\Delta(h+1)$ stands for the increment by adding node $r(h+1)$, calculated by

$$\Delta(h+1) = \sum_{j=1}^{h} A(r(j), r(h+1)). \tag{6}$$

Note that $G(1) = 0$. The above procedure can be summarized as follows.

F1. Compute $r(i)$ by sorting elements of $\mathbf{q}*$;
F2. Calculate $G(2), \cdots, G(N)$ by using (5) and (6);
F3. Output $T(h^*)$ such that $h^* = \arg\max_h G(h)$;

D. Detection algorithm

By repeatedly performing the above procedures, M times, we can detect M densely connected portions for a given network as follows.

G1. Repeat the following steps for $m = 1$ to M;
G2. Calculate $\mathbf{q}_m^*$ using **E1** to **E4**;
G3. Calculate T_m^* using **F1** to **F3**;
G4. Set $A(i,j) = 0$ if $i, j \in T_m^*$.

Here, the number M of communities is determined by a user. We estimate the mth SR-community U_m as T_m^* for any integer m with $1 \leq m \leq M$.

IV. Community Analysis of Influential Nodes

For a given network, we consider the influence maximization problem of size k under the IC model. Let $S_k = \{v_i;\ i = 1, \cdots, k\}$ be the greedy solution, and let $\mathcal{U} = \langle U_m;\ m = 1, 2, 3, \cdots \rangle$ be the SR-community structure of the network. We analyze the greedy solution S_k in terms of the SR-community structure $\mathcal{U}$.

First, we extract the minimal sequence of SR-communities in $\mathcal{U}$ such that it covers the greedy solution S_k,

$$\mathcal{U}_k = \langle U_m;\ m = 1, \cdots, M_k \rangle,$$

that is, M_k is the minimal integer M satisfying

$$\bigcup_{m=1}^{M} U_m \supset S_k.$$

Note that $\mathcal{U}_k$ can be regarded as a rough approximation to the greedy solution S_k. We call M_k the *SR-covering number* of the greedy solution S_k. For any $v_i \in S_k$, let $\alpha(v_i)$ denote the minimal integer m satisfying $U_m \ni v_i$. $U_{\alpha(v_i)}$ is reffered to as the SR-community that corresponds to the node v_i.

Next, for any $v_i \in S_k$ and a real value $p \in [0, 1]$, we consider the *influence set* $H(v_i, p)$ of v_i with probability p. Here, $H(v_i, p)$ is the set of nodes v in the network such that when $\{v_i\}$ is the initial active set, the probability that v is active at the end of the diffusion process under the IC model is more than p. Note that $v_i \in H(v_i, p) \subset H(v_i, p')$ if $0 \leq p' \leq p \leq 1$.

We investigate the correlation between the greedy solution S_k and the SR-community structure $\mathcal{U}$. In terms of F-measure, we quantify the similarity between an influence set $H(v_i, p)$ of each node v_i in the greedy solution S_k and the SR-community $U_{\alpha(v_i)}$ that correspond to v_i, that is, we measure how close the sets $H(v_i, p)$ and $U_{\alpha(v_i)}$ are by

$$F_0(p; v_i) = 200\, \frac{\left|H(v_i, p) \cap U_{\alpha(v_i)}\right|}{|H(v_i, p)| + \left|U_{\alpha(v_i)}\right|}. \tag{7}$$

Moreover, we quantify the strength of the correlation between the greedy solution S_k and the SR-community structure $\mathcal{U}$ as follows:

$$F(k) = \frac{1}{k} \sum_{i=1}^{k} F_1(v_i), \tag{8}$$

where

$$F_1(v_i) = \max_{0 \leq p \leq 1} F_0(p; v_i),\ \ (i = 1, \cdots, k).$$

V. Experimental Evaluation

Using real large networks, we experimentally evaluate the strength of the correlation between the greedy solution of the influence maximization problem under the IC model and the SR-community structure. Let $S_k = \{v_1, \cdots, v_k\}$ be the greedy solution for the influence maximization problem of size k.

A. Network Datasets

In the evaluation experiments, we should desirably use large networks that exhibit many of the key features of real social networks. Here, we report on the experimental results for two different datasets of such real networks.

First, we employed a trackback network of blogs, since a piece of information can propagate from one blog author to another blog author through a trackback. Since bloggers discuss various topics and establish mutual communications by putting trackbacks on each other's blogs, we regarded a link created by a trackback as a biderectional link for simplicity. By tracing ten steps ahead the trackbacks from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo"[1], we collected a large connected trackback network in May, 2005. This network was an undirected graph of $12,047$ nodes and $39,960$ links. This network showed the so-called "power-law" degree distribution that most real large networks exhibit. Here, the degree distribution is the distribution of the number of links for every node. We refer to this network data as *the blog network dataset*.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages. We refer to this network data as *the Wikipedia network dataset*. Here, the total numbers of nodes and links were $9,481$ and $122,522$, respectively.

Newman and Park [11] observed that social networks represented as undirected graphs generally have the following two statistical properties unlike non-social networks. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higer values of the *clustering coefficient* than the corresponding *configuration models* (i.e., random network models). Here, the clustering coefficient C for an undirected graph is defined by

$$C = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of nodes}},$$

where a "triangle" means a set of three nodes each of which is connected to each of the others, and a "connected triple" means a node connected directly to an unordered pair of others. Note that in terms of sociology, C measures the probability that two of your friends will also be friends of one another. Given a degree distribution, the corresponding configuration model of random network is defined as the ensemble of all possible graphs that possess the degree distribution, with each having equal weight. The value of C for the configuration model can be exactly calculated [10]. For the Wikipedia network, the value of C of the corresponding configuration model was 0.046, while the actual measured value of C was 0.39. Moreover, the degrees of adjacent nodes were positively correlated for the Wikipedia network dataset. Therefore, we consider that the Wikipedia network dataset can be used as an example of social network.

[1] http://blog.goo.ne.jp/usertheme/

B. A Comparison Method

In order to quantitatively evaluate the strength of the correlation between the greedy solution for the influence maximization problem under the IC model and the SR-community structure, we employ the community structure obtined by the method of Newman and Leicht [12] as a baseline.

Given an integer k, the method of Newman and Leicht [12] divides the set $S = \{1, \cdots, N\}$ of nodes in the network into k communities, that is, k disjoint subsets of S, according to some probabilistic mixture model that is a probabilistic mixture of multinomial distributions. More specifically, their method is as follows: First, a probabilistic generative model for network is given. Namely, the probability that a network with adjacency matirix $\mathbf{A}$ appears is defiend by

$$P(\mathbf{A} \,|\, \lambda, \theta) = \prod_{n=1}^{N} P(\mathbf{A}(n,:) \,|\, \lambda, \theta),$$

where $\mathbf{A}(n,:)$ denotes the nth row vector of $\mathbf{A}$,

$$\begin{aligned} \lambda &= \{\lambda_\ell;\ \ell = 1, \cdots, k\}, \\ \theta &= \{\theta_{\ell,j};\ \ell = 1, \cdots, k,\ j = 1, \cdots, N\} \end{aligned}$$

are sets of parameters, and

$$P(\mathbf{A}(n,:) \,|\, \lambda, \theta) = \sum_{\ell=1}^{k} \lambda_\ell P(\mathbf{A}(n,:) \,|\, \ell, \theta),$$

$$P(\mathbf{A}(n,:) \,|\, \ell, \theta) \propto \prod_{j=1}^{N} (\theta_{\ell,j})^{A(n,j)},$$

for $\ell = 1, \cdots, k$ and $n, j = 1, \cdots, N$. Here, each λ_ℓ is the mixture weight (prior probability) of the ℓth community, and

$$\lambda_\ell > 0,\ (\ell = 1, \cdots, k),\ \ \sum_{\ell=1}^{k} \lambda_\ell = 1.$$

Also, each $\theta_{\ell,j}$ is the probability that the jth node connects with a node beloging to the ℓth community, and

$$\theta_{\ell,j} > 0,\ \ \sum_{j=1}^{N} \theta_{\ell,j} = 1,$$

for $\ell = 1, \cdots, k$ and $j = 1, \cdots, N$. By performing the maximal likelihood estimation using the EM algorithm, we estimate the values of λ and θ. Then, applying Bayes' rule, we define the community label $\ell^*(n)$ for each node n as

$$\ell^*(n) = \arg \max_{1 \le \ell \le k} P(\ell \,|\, \mathbf{A}(n,:), \lambda, \theta).$$

For the greedy solution $S_k = \{v_1, \cdots, v_k\}$, we detect the set of k communities,

$$\mathcal{Z}_k = \{Z_1, \cdots, Z_k\},$$

by using the method of Newman and Leicht. For every v_i, we define $\gamma(v_i)$ by the condition $Z_{\gamma(v_i)} \ni v_i$. In the same way as the SR-community structure, we quantify the strength of the correlation between S_k and $\mathcal{Z}_k$ by $F(k)$ (see, (8)). Here, we modify the definition of $F(k)$ through changing each $F_0(p; v_i)$ (see, (7)) to

$$F_0(p; v_i) = 200 \frac{\left|H(v_i, p) \cap Z_{\gamma(v_i)}\right|}{|H(v_i, p)| + \left|Z_{\gamma(v_i)}\right|}.$$

C. Experimental Settings

In our experiments, we assigned a uniform probability β to the propagation probability $\beta_{u,v}$ for any directed link (u, v) of the network. As investigate by Leskovec *et al.* [7], it seems that large cascades of information diffusion happen rarely. Thus, we examined the IC model with relatively small β according to Kempe *at al.* [5].

We estimated the greedy solution $S_k = \{v_1, \cdots, v_k\}$ using the method of Kimura *et al.* [6] with the parameter value $10,000$. Here, the parameter represents the number of bond percolation processes for estimating the influence degree $\sigma(X)$ of a given initial active set X. Also, we estimated the influence set $H(v_i, p)$ of node v_i with probability p through $300,000$ simulations of the IC model.

D. Experimental Results

We describe the results for the experiments using the blog network dataset and the Wikipedia network dataset.

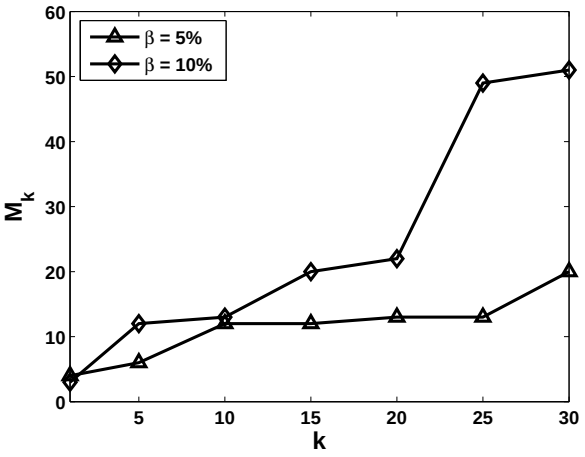


Fig. 1. SR-covering number M_k of greedy solution S_k on the blog network dataset.

Figures 1 and 2 plot the SR-covering number M_k of the greedy solution S_k with respect to k on the blog network dataset and the Wikipedia network dataset, respectively. For almost all k, we observe that the larger the value of propagation probability β is, the larger the SR-covering number M_k of S_k is.

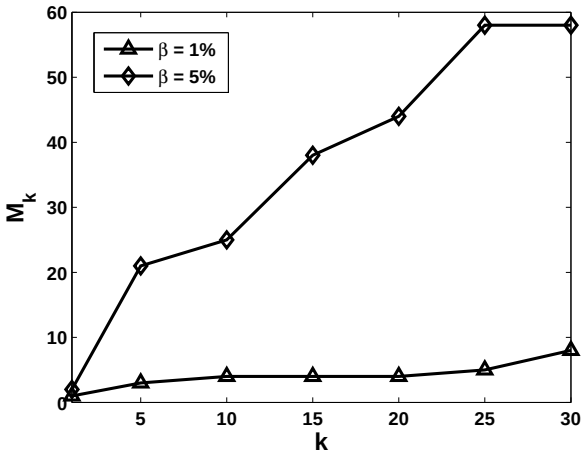


Fig. 2. SR-covering number M_k of greedy solution S_k on the Wikipedia network dataset.

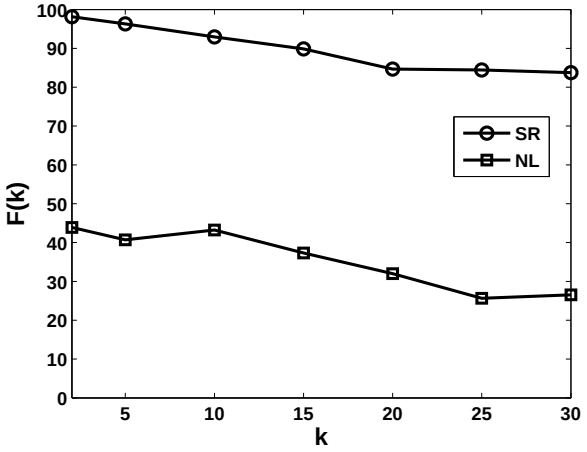


Fig. 3. Strength $F(k)$ of correlation with greedy solution S_k on the blog network dataset ($\beta = 5\%$).

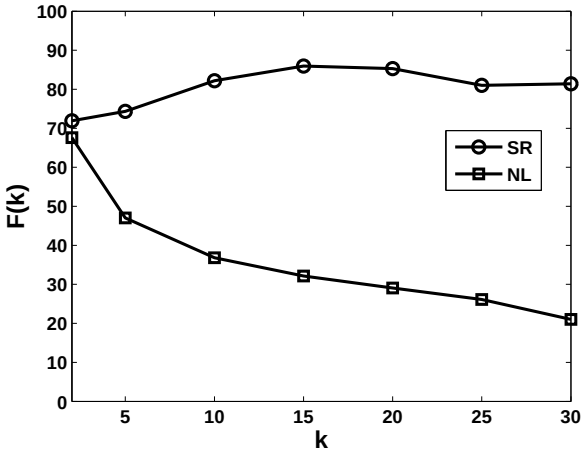


Fig. 4. Strength $F(k)$ of correlation with greedy solution S_k on the blog network dataset ($\beta = 10\%$).

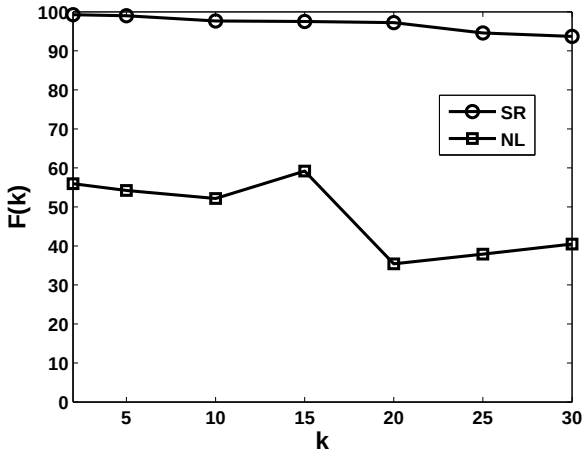


Fig. 5. Strength $F(k)$ of correlation with greedy solution S_k on the Wikipedia network dataset ($\beta = 1\%$).

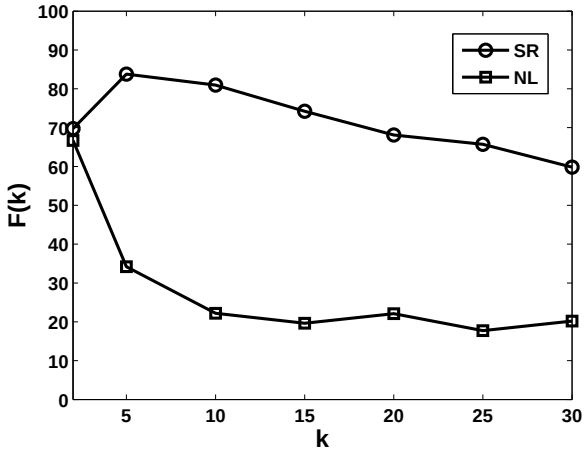


Fig. 6. Strength $F(k)$ of correlation with greedy solution S_k on the Wikipedia network dataset ($\beta = 5\%$).

Figures 3, 4, 5, and 6 plot the strength $F(k)$ of correlation with the greedy solution S_k with respect to k, $(2 \leq k \leq 30)$. In Figures 3, 4, 5, and 6, the circles indicate the strength of the correlation between the greedy solution and the SR-community structure (SR), and the squares indicate the strength of the correlation between the greedy solution and the community structure obtained by the method of Newman and Leicht (NL). Figures 3 and 4 show the results for the blog network dataset, and Figures 5 and 6 show the results for the Wikipedia network dataset. These results imply that for the IC model with relatively small propagation probability β, the SR-community structure can be more strongly correlated with the greedy solution than the community structure introduced by Newman and Leicht.

VI. Concluding Remarks

Aiming to obtain efficient methods for finding better approximate solutions for the influence maximization problem on a social network under the IC model, we have explored what structual feature of the undelying network is correlated with the greedy solution. We have focused on the SR-community structure of the network, and analyzed the greedy solution in terms of the SR-community structure. Using real large social networks including a blog network, we have experimentally demonstrated that in comparison with the community structure introduced by Newman and Leicht, the SR-community structure can be strongly correlated with the greedy solution.

On the other hand, extensive verification of this proposition with various real social networks remains an important task. However, we have already made substantial progress, and we are encouraged by our initial results.

Acknowledgment

This work was partly supported by Asian Office of Aerospace Research and Development, The U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027.

References

[1] E. Adar and L. A. Adamic, "Tracking information epidemics in blogspace," *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, Compiègne, France, 2005, pp. 207–214.

[2] P. Domingos, "Mining social networks for viral marketing," *IEEE Intelligent Systems*, vol. 20, pp. 80–82, 2005.

[3] K. J. Goldenberg, B. Libai, and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth," *Marketing Letters*, vol. 12, pp. 211–223, 2001.

[4] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, Maryland, USA, 1989.

[5] D. Kempe, J. Kleinberg, and E. Tardos, "Maximizing the spread of influence through a social network," *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, 2003, pp. 137–146.

[6] M. Kimura, K. Saito, and R. Nakano, "Extracting influential nodes for information diffusion on a social network," *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, Vancouver, British Columbia, Canada, 2007, pp. 1371-1376.

[7] J. Leskovec, A. Singh, and J. Kleinberg, "Patterns of influence in a recommendation network," *Proceedings of the 10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Singapore, 2006, pp. 380–389.

[8] A. McCallum, A. Corrada-Emmanuel, and X. Wang, "Topic and role discovery in social networks," *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, Edinburgh, Scotland, 2005, pp. 786–791.

[9] M. E. J. Newman, "The structure of scientific collaboration networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 98, pp. 404–409, 2001.

[10] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167–256, 2003.

[11] M. E. J. Newman and J. Park, "Why social networks are different from other types of networks", *Physical Review E*, vol. 68, 036122, 2003.

[12] M. E. J. Newman and E. A. Leicht, "Mixture models and exploratory analysis in networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, pp. 9564–9569, 2007.

[13] G. Palla, I. Derényi, I. Farkas, and T. Vicsek, "Uncovering the overlapping community structure of complex networks in nature and society," *Nature*, vol. 435, pp. 814–818, 2005.

[14] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing", *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Edmonton, Alberta, Canada, 2002, pp. 61–70.

[15] K. Saito, N. Ueda, M. Kimura, K. Kazama, and S. Sato, "Filtering search engine spam based on anomaly detection approach," *Proceedings of the KDD2005 Workshop on Data Mining Methods for Anomaly Detection*, Chicago, Illinois, USA, 2005, pp. 62–66.

Minimizing the Spread of Contamination by Blocking Links in a Network

Masahiro Kimura
Deptartment of Electronics and Informatics
Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics
University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research
Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract

We address the problem of minimizing the propagation of undesirable things, such as computer viruses or malicious rumors, by blocking a limited number of links in a network, a dual problem to the influence maximization problem of finding the most influential nodes in a social network for information diffusion. This minimization problem is another approach to the problem of preventing the spread of contamination by removing nodes in a network. We propose a method for efficiently finding a good approximate solution to this problem based on a naturally greedy strategy. Using large real networks, we demonstrate experimentally that the proposed method significantly outperforms conventional link-removal methods. We also show that unlike the strategy of removing nodes, blocking links between nodes with high out-degrees is not necessarily effective.

Introduction

Considerable attention has recently been devoted to investigating the structure and function of various networks including computer networks, social networks and the World Wide Web (Newman 2003). From a functional point of view, networks can mediate diffusion of various things such as innovation and topics. However, undesirable things can also spread through networks. For example, computer viruses can spread through computer networks and email networks, and malicious rumors can spread through social networks among individuals. Thus, developing effective strategies for preventing the spread of undesirable things through a network is an important research issue. Previous work studied strategies for reducing the spread size by removing nodes from a network. It has been shown in particular that the strategy of removing nodes in decreasing order of out-degree can often be effective (Albert, Jeong, and Barabási 2000; Broder et al. 2000; Callaway et al. 2000; Newman, Forrest, and Balthrop 2002). Here notice that removal of nodes by necessity involves removal of links. Namely, the task of removing links is more fundamental than that of removing nodes. Therefore, preventing the spread of undesirable things by removing links from the underlying network is an important problem.

In contrast, finding influential nodes that are effective for the spread of information through a social network is also an important research issue in terms of sociology and "viral marketing" (Domingos and Richardson 2001; Richardson and Domingos 2002; Gruhl et al. 2004). Thus, researchers (Kempe, Kleinberg, and Tardos 2003; Kimura, Saito, and Nakano 2007) have recently studied a combinatorial optimization problem called the *influence maximization problem* on a network under the *independent cascade (IC) model*, a widely-used fundamental probabilistic model of information diffusion. Here, the influence maximization problem is the problem of extracting a set of k nodes to target for initial activation such that it yields the largest expected spread of information, where k is a given positive integer. Note also that the IC model can be identified with the so-called *susceptible/infective/recoverd (SIR) model* for the spread of disease in a network (Gruhl et al. 2004).

The problem we address in this paper is a dual problem to the influence maximization problem. The problem is to minimize the spread of undesirable things by blocking a limited number of links in a network. More specifically, when some undesirable thing starts with any node and diffuses through the network under the IC model, we consider finding a set of k links such that the resulting network by blocking those links minimizes the expected contamination area of the undesirable thing, where k is a given positive integer. We refer to this combinatorial optimization problem as the *contamination minimization problem*. For this problem, we propose a novel method for efficiently finding a good approximate solution on the basis of a naturally greedy strategy. Using large real networks including a blog network, we experimentally demonstrate that the proposed method significantly outperforms link-removal heuristics that rely on the well-studied notions of betweenness and out-degree. In particular, we show that unlike the case of removing nodes, blocking links between nodes with high out-degrees is not necessarily effective for our problem.

Problem Formulation

In this paper, we address the problem of minimizing the spread of undesirable things such as computer viruses and malicious rumors in a network represented by a directed graph $G = (V, E)$. Here, V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. We

assume the IC model to be a mathematical model for the diffusion process of some undesirable thing in the network, and investigate the contamination minimization problem on G. We call nodes *active* if they have been contaminated by the undesirable thing.

Independent Cascade Model

We define the IC model on graph G according to the work of Kempe, Kleinberg, and Tardos (2003).

In the IC model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial active node v, we assume that the node v has first become active at time-step 0, and all the other nodes are inactive at time-step 0. We specify a real value p with $0 < p < 1$ in advance. Here, p is referred to as the *propagation probability* through a link. The diffusion process proceeds from the initial active node v in the following way. When a node u first becomes active at time-step t, it is given a single chance to activate each currently inactive child node w, and succeeds with probability p. If u succeeds, then w will become active at time-step $t+1$. If multiple parent nodes of w first become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate w in subsequent rounds. The process terminates if no more activations are possible.

For an initial active node v, let $\sigma(v; G)$ denote the expected number of active nodes at the end of the random process of the IC model on G. We call $\sigma(v; G)$ the *influence degree* of node v in graph G.

Contamination Minimization Problem

Now, we give a mathematical definition of the contamination minimization problem on graph $G = (V, E)$. For preventing the undesirable thing from spreading through the network under the IC model, we aim to minimize the expected contamination area (that is, the expected number of active nodes) by appropriately removing a fixed number of links.

First, we define the *contamination degree* $c(G)$ of graph G as the average of influence degrees of all the nodes in G, that is,

$$c(G) = \frac{1}{|V|} \sum_{v \in V} \sigma(v; G). \tag{1}$$

Here, $|A|$ stands for the number of elements of a set A. For any link $e \in E$, let $G(e)$ denote the graph $(V,\ E \setminus \{e\})$. We refer to $G(e)$ as the graph constructed by *blocking* e in G. Similarly, for any $D \subset E$, let $G(D)$ denote the graph $(V,\ E \setminus D)$. We refer to $G(D)$ as the graph constructed by *blocking* D in G. We define the *contamination minimization problem* on graph G as follows: Given a positive integer k with $k < |E|$, find a subset D^* of E with $|D^*| = k$ such that $c(G(D^*)) \leq c(G(D))$ for any $D \subset E$ with $|D| = k$.

For a large network, any straightforward method for exactly solving the contamination minimization problem suffers from combinatorial explosion. Therefore, we consider approximately solving the problem.

Proposed Method

We propose a method for efficiently finding a good approximate solution to the contamination minimization problem on graph $G = (V, E)$. Let k be the number of links to be blocked in this problem.

Geedy Algorithm

We approximately solve the contamination minimization problem on $G = (V, E)$ by the following greedy algorithm:

1. Set $D_0 \leftarrow \emptyset$.
2. Set $E_0 \leftarrow E$.
3. Set $G_0 \leftarrow G$.
4. **for** $i = 0$ to $k - 1$ **do**
5. Choose a link $e_* \in E_i$ minimizing $c(G_i(e))$, $(e \in E_i)$.
6. Set $D_{i+1} \leftarrow D_i \cup \{e_*\}$.
7. Set $E_{i+1} \leftarrow E_i \setminus \{e_*\}$.
8. Set $G_{i+1} \leftarrow (V, E_{i+1})$.
9. **end for**

Here, D_k is the set of links blocked, and represents the approximate solution obtained by this algorithm. G_k is the graph constructed by blocking D_k in graph G, that is, $G_k = G(D_k)$.

To implement this greedy algorithm, we need a method for calculating $\{c(G_i(e));\ e \in E_i\}$ in Step 5 of the algorithm. However, the IC model is a stochastic process model, and it is an open question to exactly calcluate influence degrees by an efficient method (Kempe, Kleinberg, and Tardos 2003). Therefore, we develop a method for estimating $\{c(G_i(e)); e \in E_i\}$.

Kimura, Saito, and Nakano (2007) presented the bond percolation method that efficiently estimates the influence degrees $\{\sigma(v; \tilde{G});\ v \in V\}$ for any directed graph $\tilde{G} = (V, \tilde{E})$. Thus, we can estimate $c(G_i(e))$ for each $e \in E_i$ by straightforwardly applying the bond percolation method. However, $|E_i|$ becomes very large for a large network unless i is very large. Therefore, we propose a method that can estimate $\{c(G_i(e));\ e \in E_i\}$ in a more efficient manner on the basis of the bond percolation method.

Bond Percolation Method

First, we revisit the bond percolation method (Kimura, Saito, and Nakano 2007). Here, we consider estimating the influence degrees $\{\sigma(v; G_i);\ v \in V\}$ for the IC model with propagation probability p in graph $G_i = (V, E_i)$.

It is known that the IC model is equivalent to the bond percolation process that independently declares every link of G_i to be "occupied" with probability p (Newman 2003). Let M be a sufficiently large positive integer. We perform the bond percolation process M times, and sample a set of M graphs constructed by the occupied links,

$$\{{G_i}^m = (V, {E_i}^m);\ m = 1, \cdots, M\}.$$

Then, we can approximate the influence degree $\sigma(v; G_i)$ by

$$\sigma(v; G_i) \simeq \frac{1}{M} \sum_{m=1}^{M} \left|\mathcal{F}(v; {G_i}^m)\right|.$$

Here, for any directed graph $\tilde{G} = (V, \tilde{E})$, $\mathcal{F}(v; \tilde{G})$ denotes the set of all the nodes that are *reachable* from node v in the graph. We say that node u is reachable from node v if there is a path from u to v along the links in the graph. Let

$$V = \bigcup_{u \in \mathcal{U}({G_i}^m)} \mathcal{S}(u; {G_i}^m)$$

be the strongly connected component (SCC) decomposition of graph ${G_i}^m$, where $\mathcal{S}(u; {G_i}^m)$ denotes the SCC of ${G_i}^m$ that contains node u, and $\mathcal{U}({G_i}^m)$ stands for a set of all the representative nodes for the SCCs of ${G_i}^m$. The bond percolation method performs the SCC decomposition of each ${G_i}^m$, and estimates all the influence degrees $\{\sigma(v; G_i); v \in V\}$ in G_i as follows:

$$\sigma(v; G_i) = \frac{1}{M} \sum_{m=1}^{M} \left| \mathcal{F}(u; {G_i}^m) \right|, \quad (v \in \mathcal{S}(u; {G_i}^m)), \quad (2)$$

where $u \in \mathcal{U}({G_i}^m)$.

Estimation Method

We are now in a position to give a method for efficiently estimating $\{c(G_i(e)); e \in E_i\}$ in Step 5 of the greedy algorithm. We develop such an estimation method on the basis of the bond percolation method.

For any directed graph $\tilde{G} = (V, \tilde{E})$, we define $\varphi(\tilde{G})$ by

$$\varphi(\tilde{G}) = \frac{1}{|V|} \sum_{v \in V} \left| \mathcal{F}(v; \tilde{G}) \right|. \quad (3)$$

Using the bond percolation method, we consider estimating the contamination degree $c(G_i)$ of the graph $G_i = (V, E_i)$. Then, by Equations (1), (2) and (3), we can estimate $c(G_i)$ as

$$c(G_i) = \frac{1}{M} \sum_{m=1}^{M} \varphi({G_i}^m). \quad (4)$$

Here, note that $\varphi({G_i}^m)$ is calculated by

$$\varphi({G_i}^m) = \frac{1}{|V|} \sum_{u \in \mathcal{U}({G_i}^m)} \left| \mathcal{F}(u; {G_i}^m) \right| \left| \mathcal{S}(u; {G_i}^m) \right|. \quad (5)$$

We assume that M is sufficiently large. Then, by the independence of the bond percolation process, we can estimate $c(G_i(e))$ for every $e \in E_i$ as

$$c(G_i(e)) = \frac{1}{|\mathcal{M}_i(e)|} \sum_{m \in \mathcal{M}_i(e)} \varphi({G_i}^m), \quad (6)$$

where ${G_i}^m = (V, {E_i}^m)$, and

$$\mathcal{M}_i(e) = \{m \in \{1, \cdots, M\};\ e \notin {E_i}^m\}.$$

We efficiently estimate $\{c(G_i(e)); e \in E_i\}$ by Equations (5) and (6) without applying the bond percolation method for every $e \in E_i$. Namely, the proposed method can achieve a great deal of reduction in computational cost compared with the coventional bond percolation method.

Experimental Evaluation

Using two large real networks, we experimentally evaluated the performance of the proposed method.

Network Datasets

First, we employed a trackback network of blogs because a piece of information can propagate from one blog author to another blog author through a trackback. Since bloggers discuss various topics and establish mutual communications by putting trackbacks on each other's blogs, we regarded a link created by a trackback as a biderectional link for simplicity. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo" (`http://blog.goo.ne.jp/usertheme/`), we collected a large connected trackback network in May, 2005. This network was a directed graph of $12,047$ nodes and $79,920$ links. We refer to this network data as the blog network.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed graph regarding those undirected links as bidirectional ones. We refer to this network data as the Wikipedia network. Here, the total numbers of nodes and directed links were $9,481$ and $245,044$, respectively.

Note here that these two networks are strongly connected.

Experimental Settings

The IC model has the propagation probability p as a parameter. So we determine the typical values of p for the blog and Wikipedia networks, and use them in the experiments. Let us consider the bond percolation process corresponding to the IC model with propagation probability p in graph $G = (V, E)$. Let S be the expected fraction of the maximal SCC in the network constructed by occupied links. S is a function of p, and as the value of p decreases, the value of S decreases. In other words, as the value of p decreases, the original graph G gradually fragments into small clusters under the corresponding bond percolation process. Figures 1 and 2 show the network fragmentation curves for the blog and Wikipedia networks, respectively. Here, we estimated S as follows:

$$S = \frac{1}{M} \sum_{m=1}^{M} \max_{u \in \mathcal{U}({G_i}^m)} |\mathcal{S}(u; {G_i}^m)|,$$

where $M = 10000$. We focus on the point p_* at which the average rate dS/dp of change of S attains the maximum, and regard it as the typical value of p for the network. Note that p_* is a critical point of dS/dp, and defines one of the features intrinsic to the network. From Figures 1 and 2, we estimated p_* to be $p_* = 0.2$ for the blog network and $p_* = 0.03$ for the Wikipedia network.

For the proposed method, we need to specify the number M of performing the bond percolation process. In the experiments, we used $M = 10000$.

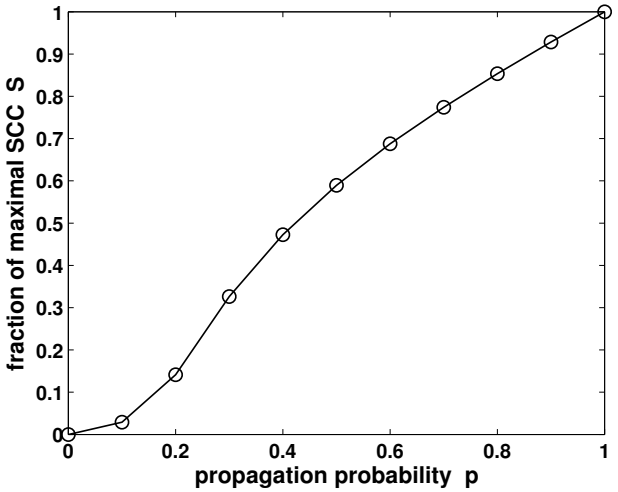


Figure 1: Fragmentation of the blog network for the IC model. The fraction S of the maximal SCC as a function of the propagation probablity p.

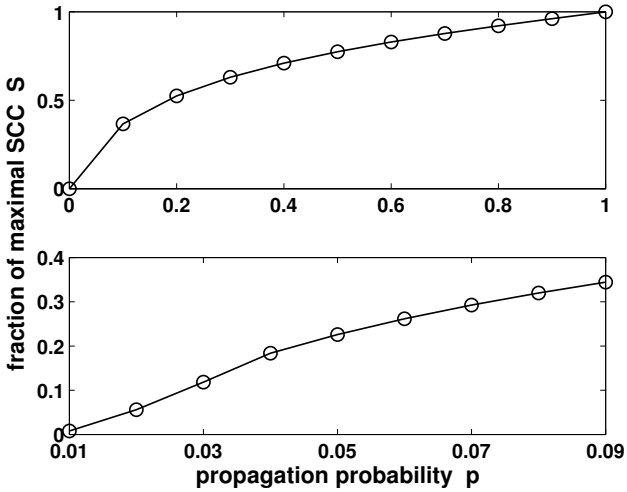


Figure 2: Fragmentation of the Wikipedia network for the IC model. The fraction S of the maximal SCC as a function of the propagation probablity p. The upper and lower frames show the network fragmentation curves for the whole range of p and the range of $0.01 \leq p \leq 0.09$, respectively.

Comparison Methods

We compared the proposed method with two heuristics based on the well-studied notions of betweenness and out-degree in the field of complex network theory, as well as the crude baseline of blocking links at random. We refer to the method of blocking links uniformly at random as the *random method*.

The *betweenness score* $b_{\tilde{G}}(e)$ of a link e in a directed graph $\tilde{G} = (V, \tilde{E})$ is defined as follows:

$$b_{\tilde{G}}(e) = \sum_{u,v \in V} \frac{n_{\tilde{G}}(e; u, v)}{N_{\tilde{G}}(u, v)},$$

where $N_{\tilde{G}}(u, v)$ denotes the number of the shortest paths from node u to node v in $\tilde{G}$, and $n_{\tilde{G}}(e; u, v)$ denotes the number of those paths that pass e. Here, we set $n_{\tilde{G}}(e; u, v)/N_{\tilde{G}}(u, v) = 0$ if $N_{\tilde{G}}(u, v) = 0$. Newman and Girvan (2004) successfully extracted community structure in a network using the following link-removal algorithm based on betweeness:

1. Calculate betweenness scores for all links in the network.
2. Find the link with the highest score and remove it from the network.
3. Recalculate betweenness scores for all remaining links.
4. Repeat from Step 2.

In particular, the notion of betweenness can be interpreted in terms of signals traveling through a network. If signals travel from source nodes to destination nodes along the shortest paths in a network, and all nodes send signals at the same constant rate to all others, then the betweenness score of a link is a measure of the rate at which signals pass along the link. Thus, we naively expect that blocking the links with the highest betweenness score can be effective for preventing the spread of contamination in the network. Therefore, we apply the method of Newman and Girvan (2004) to the contamination minimization problem. We refer to this method as the *betweenness method*.

On the other hand, previous work has shown that simply removing nodes in order of decreasing *out-degrees* works well for preventing the spread of contamination in most real networks (Albert, Jeong, and Barabási 2000; Broder et al. 2000; Callaway et al. 2000; Newman, Forrest, and Balthrop 2002). Here, the out-degree $d(v)$ of a node v means the number of outgoing links from the node v. Thus, blocking links between nodes with high out-degrees looks promising for the contamination minimization problem. Therefore, as a comparsion method, we employ the method of recursively blocking links $e = [u, v]$ from u to v in decreasing order of their scores $\overline{d}(e)$,

$$\overline{d}(e) \;=\; d(u)\, d(v).$$

We refer to this method as the *out-degree method*.

Experimental Results

We evaluated the performance of the proposed method and compare it with that of the betweenness, out-degree and random methods. Clearly, the performance of a method for solving the contamination minimization problem can be evaluated in terms of contamination degree c. We used the value of c (see Equations (4) and (5)) that is estimated by the bond percolation method with $M = 10000$.

Figures 3 and 4 show the contamination degree c of the resulting network as a function of the number k of links blocked for the blog network, where the circles, triangles, diamonds and squares indicate the results for the proposed, betweenness, out-degree and random methods, respectively.

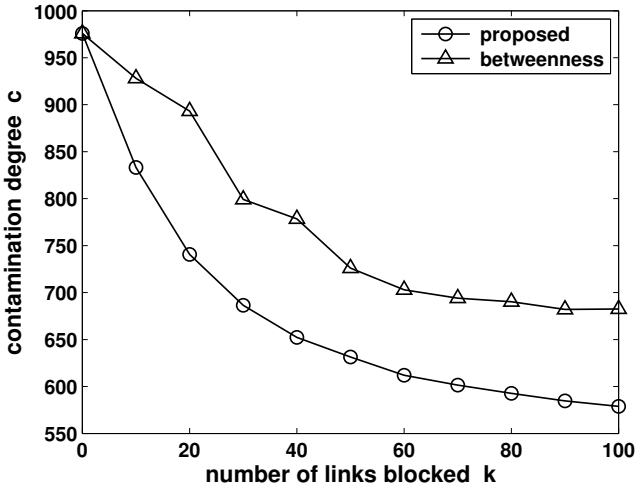


Figure 3: Performance comparison between the proposed and betweenness methods in the blog network for the IC model with $p = 0.2$.

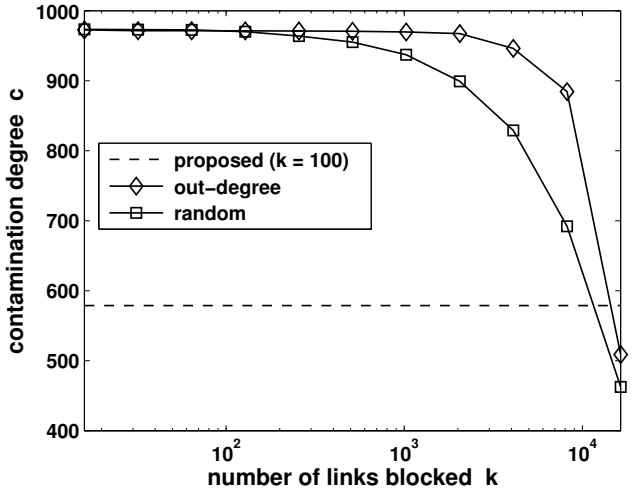


Figure 4: Performance comparison of the proposed method for $k = 100$ with the out-degree and random methods in the blog network for the IC model with $p = 0.2$.

In Figure 4, the dashed line indicates the contamination degree of the network obtained by the proposed method for $k = 100$. From Figures 3 and 4, we first see that the proposed method outperformed the betweenness, out-degree and random methods for the blog network. By taking into account the definition (1) of contamination degree, we can mention from Figure 3 that the proposed method decreased the expected number of nodes contaminated from about 980 nodes to about 580 nodes by blocking appropriate 100 links for the blog network. Here note that blocking 100 links means blocking about 0.13% of the links in the blog network. Thus, we find from Figure 3 that by appropriately blocking about 0.13% of the links in the blog network, the proposed and betweenness methods decreased contamination degree by about 41% and 30%, respectively. Hence, we can deduce that the proposed method was effective, and also outperformed the betweennes method by over 10% at $k = 100$ for the blog network. Moreover, we find from Figure 4 that blocking 100 links by using the proposed method was the same as blocking over 10000 links by using the out-degree and random methods for the blog network in effect. Namely, we can deduce that the proposed method was 100 times more effective than the out-degree and random methods at $k = 100$ for the blog network.

Figures 5 and 6 display the contamination degree c of the resulting network as a function of the number k of links blocked for the Wikipedia network. Here, as in Figures 3 and 4, the circles, triangles, diamonds and squares indicate the results for the proposed, betweenness, out-degree and random methods, respectively. In Figure 6, the dashed line indicates the contamination degree of the network obtained by the proposed method for $k = 300$. We also see from Figures 5 and 6 that the proposed method outperformed the betweenness, out-degree and random methods for the Wikipedia network. In particualr, we observe from Figure 5 that as the value of k increased, the performance difference between the proposed and betweenness methods gradually increased. Note here that blocking 300 links means blocking about 0.12% of the links in the Wikipedia network. Thus, we find from Figure 5 that by appropriately blocking about 0.12% of the links in the Wikipedia network, the proposed and betweenness methods decreased contamination degree by about 26% and 16%, respectively. Hence, we can deduce that the proposed method was effective, and also outperformed the betweennes method by about 10% at $k = 300$ for the Wikipedia network. Moreover, we find from Figure 6 that blocking 300 links by using the proposed method was the same as blocking about 30000 links by using the out-degree and random methods for the Wikipedia network. Namely, we can deduce that the proposed method was effective about 100 times as much as the out-degree and random methods at $k = 300$ for the Wikipedia network.

These results imply that the proposed method works effectively as expected, and significantly outperforms the conventional link-removal heuristics, that is, the betweenness, out-degree and random methods. This shows that a significantly better link-blocking strategy for reducing the spread size of contamination can be obtained by explicitly incorporating the diffusion dynamics of contamination in a network, rather than relying solely on structual properties of the graph.

We note from Figures 4 and 6 that the out-degree method was almost the same as or worse than the random method in performance. In the task of removing nodes from a network, the out-degree heuristic has been effective since many links can be blocked at the same time by removing nodes with high out-degrees. However, we find that in the task of blocking a limited number of links, the strategy of blocking links between nodes with high out-degrees is not necessarily effective.

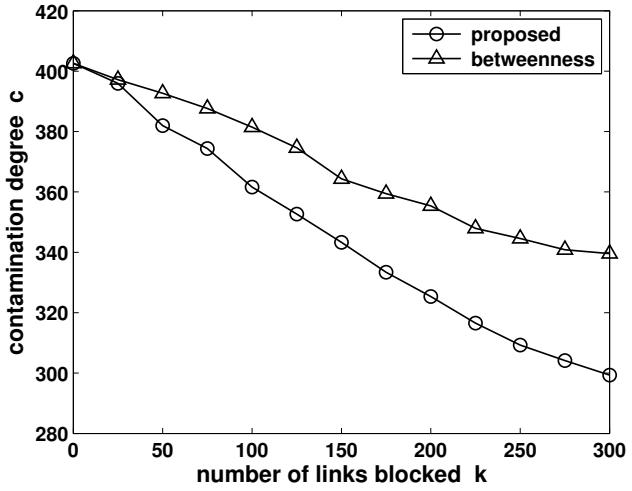


Figure 5: Performance comparison between the proposed and betweenness methods in the Wikipedia network for the IC model with $p = 0.03$.

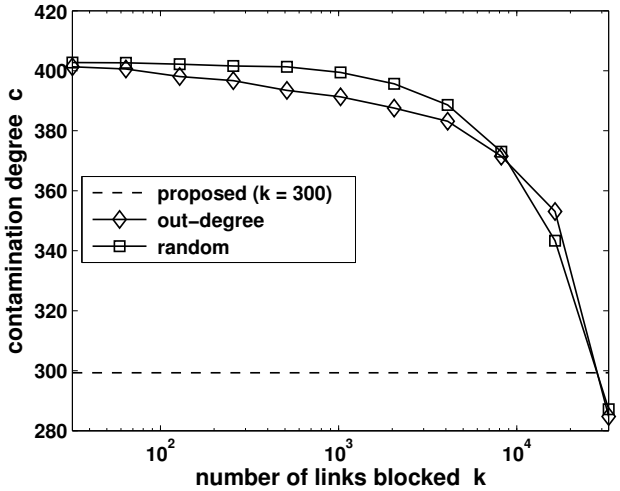


Figure 6: Performance comparison of the proposed method for $k = 300$ with the out-degree and random methods in the Wikipedia network for the IC model with $p = 0.03$.

Conclusion

In an attempt to minimize the spread of undesirable things by blocking links in a network, we have considered the contamination minimization problem, a dual problem to the influence maximization problem for social networks. This minimization problem is another approach to the problem of preventing the spread of contamination by removing nodes in a network, We have proposed a novel method for efficiently finding a good approximate solution to this problem on the basis of the greedy algorithm and the bond percolation method. Using large-scale blog and Wikipedia networks, we have experimentally demonstrated that the proposed method effectively works, and also significantly outperforms the conventional link-removal heuristics based on the betweenness and out-degree. Moreover, we have found that unlike the task of removing nodes, the strategy of blocking links between nodes with high out-degrees is not necessarily effective for our problem.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027.

References

Albert, R.; Jeong, H.; and Barabási, A.-L. 2000. Error and attack tolerance of complex networks. *Nature* 406:378–382.

Broder, A.; Kumar, R.; Maghoul, F.; Raghavan, P.; Rajagopalan, S.; Stata, R.; Tomkins, A.; and Wiener, J. 2000. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference*, 309–320.

Callaway, D. S.; Newman, M. E. J.; Strogatz, S. H.; and Watts, D. J. 2000. Network robustness and fragility: Percolation on random graphs. *Physical Reveiw Letters* 85:5468–5471.

Domingos, P., and Richardson, M. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 57–66.

Gruhl, D.; Guha, R.; Liben-Nowell, D.; and Tomkins, A. 2004. Information diffusion through blogspace. In *Proceedings of the 13th International World Wide Web Conference*, 107–117.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 137–146.

Kimura, M.; Saito, K.; and Nakano, R. 2007. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 1371–1376.

Newman, M. E. J., and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E* 69:026113.

Newman, M. E. J.; Forrest, S.; and Balthrop, J. 2002. Email networks and the spread of computer viruses. *Physical Review E* 66:035101.

Newman, M. E. J. 2003. The structure and function of complex networks. *SIAM Review* 45:167–256.

Richardson, M., and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 61–70.

Effective Visualization of Information Diffusion Process over Complex Networks

Kazumi Saito[1], Masahiro Kimura[2], and Hiroshi Motoda[3]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. Effective visualization is vital for understanding a complex network, in particular its dynamical aspect such as information diffusion process. Existing node embedding methods are all based solely on the network topology and sometimes produce counter-intuitive visualization. A new node embedding method based on conditional probability is proposed that explicitly addresses diffusion process using either the IC or LT models as a cross-entropy minimization problem, together with two label assignment strategies that can be simultaneously adopted. Numerical experiments were performed on two large real networks, one represented by a directed graph and the other by an undirected graph. The results clearly demonstrate the advantage of the proposed methods over conventional spring model and topology-based cross-entropy methods, especially for the case of directed networks.

1 Introduction

Analysis of the structure and function of complex networks, such as social, computer and biochemical networks, has been a hot research subject with considerable attention [10]. A network can play an important role as a medium for the spread of various information. For example, innovation, hot topics and even malicious rumors can propagate through social networks among individuals, and computer viruses can diffuse through email networks. Previous work addressed the problem of tracking the propagation patterns of topics through network spaces [5, 1], and studied effective "vaccination" strategies for preventing the spread of computer viruses through networks [11, 2]. Widely-used fundamental probabilistic models of information diffusion through networks are the *independent cascade (IC) model* and the *linear threshold (LT) model* [8, 5]. Researchers have recently investigated the problem of finding a limited number of influential nodes that are effective for the spread of information through a network under these models [8, 9]. In these studies, understanding the flow of information through networks is an important research issue.

This paper focuses on the problem of visualizing the information diffusion process, which is vital for understanding its characteristic over a complex network. Existing node embedding methods such as spring model method [7] and cross entropy method [14] are solely based on the network topology. They do not take account how information diffuses across the network. Thus, it often happens that the visualized information flow do not match our intuitive understanding, *e.g.*, abrupt information flow gaps, inconsistency between the nodes distance and the reachability of information, irregular pattern of information spread, etc. This sometimes happens when visualizing the diffusion process for a network represented by a directed graph.

Thus, it is important that node embedding explicitly reflects the diffusion process to produce more natural visualization. We have devised a new node embedding method that incorporates conditional probability of information diffusion between two nodes, a target source node where the information is initially issued and a non-target influenced node where the information has been received via intermediate nodes. Our postulation is that good visualization should satisfy the two conditions: path continuity, *i.e.* any information diffusion path is continuous and path separability, *i.e.* each different information diffusion path is clearly separated from each other. To this end, the above node embedding is coupled with two label assignment strategies, one with emphasis on influence of initially activated nodes, and the other on degree of information reachability.

Extensive numerical experiments were performed on two large real networks, one generated from a large connected trackback network of blog data, resulting in a directed graph of $12,047$ nodes and $53,315$ links, and the other, a network of people, generated from a list of people within a Japanese Wikipedia, resulting in an undirected graph of $9,481$ nodes and $245,044$ links. The results clearly indicate that the proposed probabilistic visualization method satisfies the above two conditions and demonstrate its advantage over the well-known conventional methods: spring model and topology-based cross-entropy methods, especially for the case of a directed network. The method appeals well to our intuitive understanding of information diffusion process.

2 Information Diffusion Models

We mathematically model the spread of information through a directed network $G = (V, E)$ under the IC or LT model, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively. We call nodes *active* if they have been influenced with the information. In these models, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set S of active nodes, we assume that the nodes in S have first become active at time-step 0, and all the other nodes are inactive at time-step 0.

2.1 Independent Cascade Model

We define the IC model. In this model, for each directed link (u, v), we specify a real value $\beta_{u,v}$ with $0 < \beta_{u,v} < 1$ in advance. Here $\beta_{u,v}$ is referred to as the *propagation probability* through link (u, v). The diffusion process proceeds from a given initial active

set S in the following way. When a node u first becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\beta_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. If multiple parent nodes of v first become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set S, let $\varphi(S)$ denote the number of active nodes at the end of the random process for the IC model. Note that $\varphi(S)$ is a random variable. Let $\sigma(S)$ denote the expected value of $\varphi(S)$. We call $\sigma(S)$ the *influence degree* of S.

2.2 Linear Threshold Model

We define the LT model. In this model, for every node $v \in V$, we specify, in advance, a *weight* $\omega_{u,v}$ (> 0) from its parent node u such that

$$\sum_{u \in \Gamma(v)} \omega_{u,v} \leq 1,$$

where $\Gamma(v) = \{u \in V; (u,v) \in E\}$. The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0,1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is,

$$\sum_{u \in \Gamma_t(v)} \omega_{u,v} \geq \theta_v,$$

then v will become active at time-step $t+1$. Here, $\Gamma_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

The LT model is also a probabilistic model associated with the uniform distribution on $[0,1]^{|V|}$. Similarly to the IC model, we define a random variable $\varphi(S)$ and its expected value $\sigma(S)$ for the LT model.

2.3 Influence Maximization Problem

Let K be a given positive integer with $K < |V|$. We consider the problem of finding a set of K nodes to target for initial activation such that it yields the largest expected spread of information through network G under the IC or LT model. The problem is referred to as the *influence maximization problem*, and mathematically defined as follows: Find a subset S^* of V with $|S^*| = K$ such that $\sigma(S^*) \geq \sigma(S)$ for every $S \subset V$ with $|S| = K$.

For a large network, any straightforward method for exactly solving the influence maximization problem suffers from combinatorial explosion. Therefore, we approximately solve this problem. Here, $U_K = \{u_1, \cdots, u_K\}$ is the set of K nodes to target for initial activation, and represents the approximate solution obtained by this algorithm. We refer to U_K as the *greedy solution*.

Using large collaboration networks, Kempe et al. [8] experimentally demonstrated that the greedy algorithm significantly outperforms node-selection heuristics that rely on the well-studied notions of degree centrality and distance centrality in the sociology literature. Moreover, the quality of U_K is guaranteed:

$$\sigma(U_K) \geq \left(1 - \frac{1}{e}\right) \sigma(S_K^*),$$

where S_K^* stands for the exact solution to this problem.

To implement the greedy algorithm, we need a method for calculating $\{\sigma(U_k \cup \{v\}); v \in V \setminus U_k\}$ for $1 \leq k \leq K$. However, it is an open question to exactly calculate influence degrees by an efficient method for the IC or LT model [8]. Kimura et al. [9] presented the bond percolation method that efficiently estimates influence degrees $\{\sigma(U_k \cup \{v\}); v \in V \setminus U_k\}$. Therefore, we estimate the greedy solution U_K using their method.

3 Visualization Method

We especially focus on visualizing the information diffusion process from the target nodes selected to be a solution of the influence maximization problem. To this end, we propose a visualization method that has the following characteristics: 1) utilizing the target nodes as a set of pivot objects for visualization, 2) applying a probabilistic algorithm for embedding all the nodes in the networks into an Euclidean space, and 3) varying appearance of the embedded nodes on the basis of two label assignment strategies. In what follows, we describe some details of the probabilistic embedding algorithm and the label assignment strategies.

3.1 Probabilistic Embedding Algorithm

Let $U_K = \{u_k : 1 \leq k \leq K\} \subset V$ be a set of target nodes, which maximizes an expected number of influenced nodes in the network based on an information diffusion model such as IC or LT. Let $v_n \notin U_K$ be a non-target node in the network, then we can consider the conditional probability $p_{k,n} = p(v_n|u_k)$ that a node v_n is influenced when one target node u_k alone is set to an initial information source. Here note that we can regard $p_{k,n}$ as a binomial probability with respect to a pair of nodes u_k and v_n. In our visualization approach, we attempt to produce embedding of the nodes so as to preserve the relationships expressed as the conditional probabilities for all pairs of target and non-target nodes in the network. We refer to this visualization strategy as the *conditional probability embedding (CE) algorithm*.

Objective Function Let $\{\mathbf{x}_k : 1 \leq k \leq K\}$ and $\{\mathbf{y}_n : 1 \leq n \leq N\}$ be the embedding positions of the corresponding K target nodes and $N = |V| - K$ non-target nodes in an M dimensional Euclidean space. Hereafter, the $\mathbf{x}_k$ and $\mathbf{y}_n$ are called target and non-target vectors, respectively. As usual, we define the Euclidean distance between $\mathbf{x}_k$ and $\mathbf{y}_n$ as follows:

$$d_{k,n} = \|\mathbf{x}_k - \mathbf{y}_n\|^2 = \sum_{m=1}^{M} (x_{k,m} - y_{n,m})^2.$$

Here, we introduce a monotonic decreasing function $\rho(s) \in [0, 1]$ with respect to $s \geq 0$, where $\rho(0) = 1$ and $\rho(\infty) = 0$.

Since $\rho(d_{k,n})$ can also be regarded as a binomial probability with respect to $\mathbf{x}_k$ and $\mathbf{y}_n$, we can introduce a cross-entropy (cost) function between $p_{k,n}$ and $\rho(d_{k,n})$ as follows:

$$\mathcal{E}_{k,n} = -p_{k,n} \ln \rho(d_{k,n}) - (1 - p_{k,n}) \ln(1 - \rho(d_{k,n})).$$

Since $\mathcal{E}_{k,n}$ is minimized when $\rho(d_{k,n}) = p_{k,n}$, this minimization with respect to $\mathbf{x}_k$ and $\mathbf{y}_n$ is consistent with our problem setting. In this paper, we employ a function of the form

$$\rho(s) = \exp\left(-\frac{s}{2}\right)$$

as the monotonic decreasing function, but note that our approach is not restricted to this form. Then, the total cost function (objective function) can be defined as follows:

$$\mathcal{E} = \frac{1}{2} \sum_{n=1}^{N} \sum_{k=1}^{K} p_{k,n} d_{k,n} - \sum_{n=1}^{N} \sum_{k=1}^{K} (1 - p_{k,n}) \ln(1 - \rho(d_{k,n})). \tag{1}$$

Namely, our approach is formalized as a minimization problem of the objective function defined in (1) with respect to $\{\mathbf{x}_k : 1 \leq k \leq K\}$ and $\{\mathbf{y}_n : 1 \leq n \leq N\}$.

Learning Algorithm As the basic structure of our learning algorithms, we adopt a coordinate strategy just like the *EM* (Expectation-Maximization) algorithm. First, we adjust the target vectors, so as to minimize the objective function by freezing the non-target vectors, and then, we adjust the non-target vectors by freezing the target vectors. These two steps are repeated until convergence is obtained.

In the former minimization step for the *CE* algorithm, we need to calculate the derivative of the objective function with respect to $\mathbf{x}_n$ as follows:

$$\mathcal{E}_{\mathbf{x}_k} = \frac{\partial \mathcal{E}}{\partial \mathbf{x}_k} = \sum_{n=1}^{N} \frac{p_{k,n} - \rho(d_{k,n})}{1 - \rho(d_{k,n})} (\mathbf{x}_k - \mathbf{y}_n). \tag{2}$$

Since $\mathbf{x}_{k'}$ $(k' \neq k)$ disappears in (2), we can update $\mathbf{x}_k$ without considering the other target vectors. In the latter minimization step for the *CE* algorithm, we need to calculate the following derivative,

$$\mathcal{E}_{\mathbf{y}_n} = \frac{\partial \mathcal{E}}{\partial \mathbf{y}_n} = \sum_{k=1}^{K} \frac{p_{k,n} - \rho(d_{k,n})}{1 - \rho(d_{k,n})} (\mathbf{y}_n - \mathbf{x}_k).$$

In this case, we update $\mathbf{y}_n$ by freezing the other non-target vectors. Overall, our algorithm can be summarized as follows:

1. Initialize vectors $\mathbf{x}_1, \cdots, \mathbf{x}_K$ and $\mathbf{y}_1, \cdots, \mathbf{y}_N$.
2. Calculate gradient vectors $\mathcal{E}_{\mathbf{x}_1}, \cdots, \mathcal{E}_{\mathbf{x}_K}$.
3. Update target vectors $\mathbf{x}_1, \cdots, \mathbf{x}_K$.
4. Calculate gradient vectors $\mathcal{E}_{\mathbf{y}_1}, \cdots, \mathcal{E}_{\mathbf{y}_N}$.

5. Update non-target vectors $\mathbf{y}_1, \cdots, \mathbf{y}_N$.
6. Stop if $\max_{k,n}\{\|\mathcal{E}_{\mathbf{x}_k}\|, \|\mathcal{E}_{\mathbf{y}_n}\|\} < \epsilon$.
7. Return to **2**.

Here, a small positive value ϵ controls the termination condition.

3.2 Label Assignment Strategies

In an attempt to effectively understand information diffusion process, we propose two label assignment strategies, on which the appearance of the embedded target and non-target nodes depends. The first strategy assigns labels to non-target nodes according to the standard Bayes decision rule.

$$l_1(v_n) = \arg \max_{1 \leq k \leq K} \{p_{k,n}\}$$

It is obvious that this decision naturally reflects influence of the target nodes. Note that the target node identification number k corresponds to the order determined by the greedy method, *i.e.*, $l_1(u_k) = k$.

In the second strategy, we introduce the following probability quantization by noting $0 \leq \max_{1 \leq k \leq K}\{p_{k,n}\} \leq 1$,

$$l_2(v_n) = \left[-\log_b \max_{1 \leq k \leq K} \{p_{k,n}\} \right] + 1,$$

where $[x]$ returns the greatest integer not greater than x, and b stands for the base of logarithm. To each node belonging to $Z = \{v_n : \max_{1 \leq k \leq K}\{p_{k,n}\} = 0\}$, we assign as the label the maximum number determined by the nodes not belonging to Z. We believe that this quantization reasonably reflects the degree of information reachability. Here note that $l_2(u_k) = 1$ because it always becomes active at time step $t = 0$. These labels are further mapped to colors scales according to some monotonic mapping functions.

4 Experimental Evaluation

4.1 Network Data

In our experiments, we employed two sets of real networks used in [9], which exhibit many of the key features of social networks. We describe the details of these network data.

The first one is a trackback network of blogs. Blogs are personal on-line diaries managed by easy-to-use software packages, and have rapidly spread through the World Wide Web [5]. Bloggers (*i.e.*, blog authors) discuss various topics by using trackbacks. Thus, a piece of information can propagate from one blogger to another blogger through a trackback. We exploited the blog "Theme salon of blogs" in the site "goo" [2], where a blogger can recruit trackbacks of other bloggers by registering an interesting theme. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR

[2] `http://blog.goo.ne.jp/usertheme/`

Fukuchiyama Line Derailment Collision", we collected a large connected trackback network in May, 2005. The resulting network had 12,047 nodes and 53,315 directed links, which features the so-called "power-law" distributions for the out-degree and in-degree that most real large networks exhibit. We refer to this network data as the blog network.

The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages. The undirected graph is represented by an equivalent directed graph by regarding undirected links as bidirectional ones[3]. The resulting network had 9,481 nodes and 245,044 directed links. We refer to this network data as the Wikipedia network.

Newman and Park [12] observed that social networks represented as undirected graphs generally have the following two statistical properties that are different from non-social networks. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* C than the corresponding *configuration models* (*i.e.*, random network models). For the undirected graph of the Wikipedia network, the value of C of the corresponding configuration model was 0.046, while the actual measured value of C was 0.39, and the degrees of adjacent nodes were positively correlated. Therefore, the Wikipedia network has the key features of social networks.

4.2 Experimental Settings

In the IC model, we assigned a uniform probability β to the propagation probability $\beta_{u,v}$ for any directed link (u, v) of a network, that is, $\beta_{u,v} = \beta$. We, first, determine the typical value of β for the blog network, and use it in the experiments. It is known that the IC model is equivalent to the bond percolation process that independently declares every link of the network to be "occupied" with probability β [10]. Let J denote the expected fraction of the maximal strongly connected component (SCC) in the network constructed by the occupied links. Note that J is an increasing function of β. We focus on the point β_* at which the average rate of change of J, $dJ/d\beta$, attains the maximum, and regard it as the typical value of β for the network. Note that β_* is a critical point of $dJ/d\beta$, and defines one of the features intrinsic to the network. Figure 1 plots J as a function of β. Here, we estimated J using the bond percolation method with the same parameter value as below [9]. From this figure we experimentally estimated β_* to be 0.2 for the blog network. In the same way, we experimentally estimated β_* to be 0.05 for the Wikipedia network.

In the LT model, we uniformly set weights as follows. For any node v of a network, the weight $\omega_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by $\omega_{u,v} = 1/|\Gamma(v)|$.

Once these parameters were set, we estimated the greedy solution $U_K = \{u_1, \cdots, u_K\}$ of targets and the conditional probabilities $\{p_{k,n};\ 1 \le k \le K,\ 1 \le n \le N\}$ using the bond percolation method with the parameter value 10,000 [9]. Here, the parameter represents

[3] For simplicity, we call a graph with bi-directional links an undirected graph

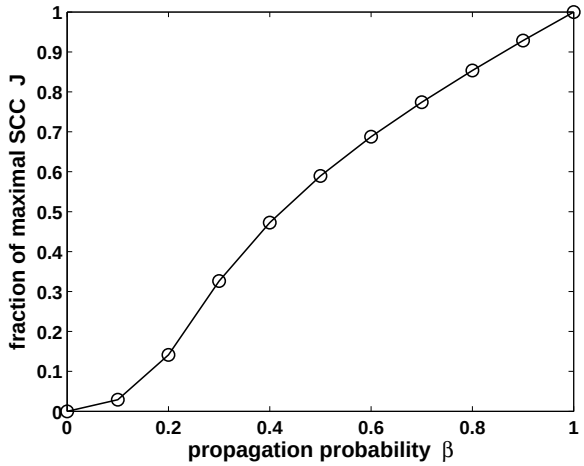


Fig. 1: The fraction J of the maximal SCC as a function of the propagation probability β

the number of bond percolation processes for estimating the influence degree $\sigma(S)$ of a given initial active set S.

4.3 Brief Description of Other Visualization Methods used for Comparison

We have compared the proposed method with the two well known methods: spring model method [7] and standard cross-entropy method [14].

Spring model method assumes that there is a hypothetical spring between each connected node pair and locates nodes such that the distance of each node pair is closest to its minimum path length at equilibrium. Mathematically it is formulated as minimizing (3).

$$\mathcal{K}(\mathbf{x}) = \sum_{u=1}^{|V|-1} \sum_{v=u+1}^{|V|} \alpha_{u,v} (g_{u,v} - \|\mathbf{x}_u - \mathbf{x}_v\|)^2, \tag{3}$$

where $g_{u,v}$ is the minimum path length between node u and node v, and $\alpha_{u,v}$ is a spring constant which is normally set to $1/(2g_{u,v}^2)$. Standard cross-entropy method first defines a similarity $\rho(\|\mathbf{x}_u - \mathbf{x}_v\|^2) = \exp(-\|\mathbf{x}_u - \mathbf{x}_v\|^2/2)$ between the embedding coordinates x_u and x_v and uses the corresponding element $a_{u,v}$ of the adjacency matrix as a measure of distance between the node pair, and tries to minimize the total cross entropy between these two. Mathematically it is formulated as minimizing (4).

$$C(\mathbf{x}) = \sum_{u=1}^{|V|-1} \sum_{v=u+1}^{|V|} \left\{ -a_{u,v} \log \rho(\|\mathbf{x}_u - \mathbf{x}_v\|^2) - (1 - a_{u,v}) \log(1 - \rho(\|\mathbf{x}_u - \mathbf{x}_v\|^2)) \right\}, \tag{4}$$

Here, note that we used the same function ρ as before.

As is clear from the above formulation, both methods are completely based on graph topology. They are both non-linear optimization problem and easily solved by a standard coordinate descent method. Here note that the applicability of the spring model method and cross-entropy method is basically limitted to undirected networks. Thus, in order to obtain the embedding results by using these methods we neglected the direction in the directed blog network and regarded it as undirected one.

4.4 Experimental Results

Two label assignment strategies are independent to each other. They can be used either separately or simultaneously. Here, we used a color mapping to both, and thus, use them separately. The visualization results are shown in four figures, each with six network figures. In each of these four figures, the left three show the results of the first visualization strategy (method 1) and the right three the results of the second visualization strategy (method 2), and the top two show the results of the proposed method (*CE* algorithm), the middle two the results of spring model and the bottom two the results of the topology-based cross entropy method. The first two figures (Figs. 2 and 3) corresponds to the results of blog network and the last two (Figs. 4 and 5) the results of Wikipedia network. For each, the results of the IC model comes first, followed by the results of the LT model.

The most influential top ten nodes are chosen as the target nodes, and the rest are all non-target nodes. In the first visualization strategy, the color of a non-target node indicates which target node is most influential to the node, whereas in the second visualization strategy, it indicates how easily the information diffuses from the most influential target node to reach the node. Note that a non-target node is influenced by multiple target nodes probabilistically, but here the target with the highest conditional probability is chosen. Thus, the most influential target node is determined for each non-target node.

Observation of the results of the proposed method (Figs. 2a, 2b, 3a, 3b, 4a, 4b, 5a, and 5b) indicates that the proposed method has the following desirable characteristics: 1) the target nodes tend to be allocated separately from each other, and from each target node, 2) the non-target nodes that are most affected by the same target node are laid out forming a band and 3) the reachability changes continuously from the highest at the target node to the lowest at the other end of the band. From this observation, it is confirmed that the two conditions we postulated are satisfied for the both diffusion models. Observation 2) above, however, needs further clarification. Note that our visualization does not necessarily cause the information diffusion to neighboring nodes to be in the form of a line in the embedded space. For example, if there is only one source (K=1), the information would diffuse concentrically. A node in general receives information from multiple sources. The fact that the embedding result forms a line, on the contrary, reveals an important characteristic that little information is coming from the other sources for the networks we analyzed.

In the proposed method, non-target nodes that are readily influenced are easily identified, whereas those that are rarely influenced are placed together. Overlapping of the color well explains the relationship between each target and a non-target node. For example, in Figures 3a and 3b it is easily observed that the effect of the target nodes

5, 2 on non-target nodes interferes with the three bands that are spread from the target nodes 8, 3, 10, and non-target nodes overlap as they move away from the target nodes, demonstrating that a simple two-dimensional visualization facilitates how different node groups overlap and how the information flows from different target nodes interfere each other. The same observation applies for the target nodes 6, 1, 9, 7. On the contrary, the target node 4 has its own effect separately. A similar argument is possible for relationship within target nodes. For example, in Figures 2a target nodes 4, 5, 6, 8 are located in relatively near positions compared with the other target nodes. It is crucial to abstract and visualize the essence of information diffusion by deleting the unnecessary details (node to node diffusion). A good explanation for the overlap like the above is not possible by other visualization methods. Further, the visualization results of both IC and LT models are spatially well balanced. In addition, there are no significant differences on the results of visualization between the directed network and undirected network. Both are equally good.

Observation of the results of the spring model (Figs. 2c, 2d, 3c, 3d, 4c, 4d, 5c, and 5d) and the topology-based cross entropy method (Figs. 2e, 2f, 3e, 3f, 4e, 4f, 5e, and 5f) reveals the followings. The clear difference of these from the proposed method is that it is not that easy to locate the target nodes. This is true, in particular, for the spring model. It is slightly easier for the standard cross-entropy method because the target nodes are placed in the cluster centers, but clusters often overlap, which makes visualization less understandable. It is also noted that those nodes with high reachability, *i.e.*, nodes with red, which should be placed separately due to the influence of different target nodes are placed in mixture. Further, unlike the proposed method, there is clear difference between the IC model and the LT model. In the IC model, we can easily recognize non-target nodes with high reachability, which cover a large portion of the network, whereas in the LT model, such nodes covering only a small portion are almost invisible in the network. In contrast, we can easily pick up such non-target nodes with high reachability even for the LT model in the proposed method.

We observe that the standard cross-entropy method is in general better than the spring model method in terms of the clarity of separability. The standard cross-entropy method does better for the IC model than for the LT model, and is comparable to the proposed method in terms of the clarity of reachability. However, the results of the standard cross-entropy method (e.g., Fig. 2f) are unintuitive, where the high reachability non-target nodes are placed away from the target nodes, and some target node forms several isolated clusters. We believe that this point is an intrinsic limitation of the standard cross-entropy method.

The concept of our visualization is based on the notion that how the information diffuses should primarily determine how the visualization is made, irrespective of the graph topology. We observe that the visualization which is based solely on the topology has intrinsic limitation when we deal with a huge network from the point of both computational complexity (*e.g.*, the spring model does not work for a network with millions nodes) and understandability. Overall, we can conclude that the proposed method provides better visualization which is more intuitive and easily understandable.

5 Related Work and Discussion

As defined earlier, let K and N be the numbers of target and non-target nodes in a network. Then the computational complexity of our embedding method amounts to $O(NK)$, where we assume the number of learning iterations and the embedding dimension to be constants. This reduced complexity greatly expands the applicability of our method over the other representative network embedding methods, *e.g.*, the spring model method [7] and the standard cross-entropy method [14], both of which require the computational complexity of $O(N^2)$ under the setting that $K \ll N$.

In view of computational complexity, our visualization method is closely related to those conventional methods, such as FastMap or Landmark Multidimensional Scaling (LMDS), which are based on the Nyström approximation [13]. Typically, these methods randomly select a set of pivot (or landmark) objects, then produce the embedding results so as to preserve relationships between all pairs of pivot and non-pivot objects. In contrast, our method selects target (pivot) nodes based on the information diffusion models.

Our method adopts the basic idea of the probabilistic embedding algorithms including Parametric Embedding (PE) [6] and Neural Gas Cross-Entropy (NG-CE) [4]. The PE method attempts to uncover classification structures by use of posterior probabilities, while the NG-CE method is restricted to visualize the codebooks of the neural gas model. Our purpose, on the other hand, is to effectively visualize information diffusion process. The two visualization strategies we proposed match this aim.

We are not the first to try to visualize the information diffusion process. Adar and Adamic [1] presented a visualization system that tracks the flow of URL through blogs. However, same as above, their visualization method did not incorporate an information diffusion model. Further, they laid out only a small number of nodes in a tree structure, and it is unlikely that their approach scales up to a large network.

Finally we should emphasize that unlike most representative embedding methods for networks [3], our visualization method is applicable to large-scale directed graphs while incorporating the effect of information diffusion models. In this paper, however, we also performed our experiments using the undirected (bi-directional) Wikipedia network. This is because we wanted to include favorable evaluation for the comparison methods. As noted earlier, we cannot directly apply the conventional embedding methods to directed graphs without some topology modification such as link addition or deletion.

6 Conclusion

We proposed an innovative probabilistic visualization method to help understand complex network. The node embedding scheme in the method, formulated as a model-based cross-entropy minimization problem, explicitly take account of information diffusion process, and therfore, the resulting visualization is more intuitive and easier to understand than the state-of-art approaches such as the spring model method and the standard cross-entropy method. Our method is efficient enough to be applied to large networks. The experiments performed on a large blog network (directed) and a large Wikipedia

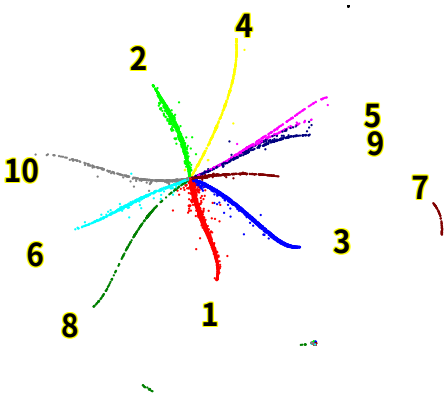
network (undirected) clearly demonstrate the advantage of the proposed method. The proposed method is confirmed to satisfy both path continuity and path separability conditions which are the important requirement for the visualization to be understandable. Our future work includes the extension of the proposed approach to the visualization of growing networks.

Acknowledgments

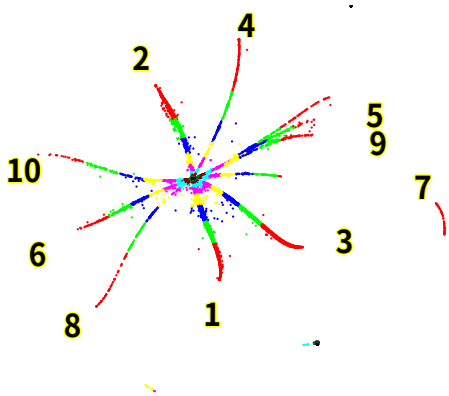
This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

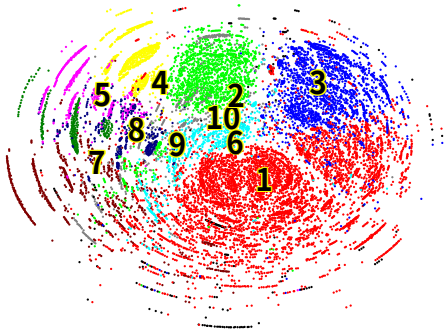
1. Adar, E., & Adamic, L. (2005). Tracking information epidemics in blogspace. *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 207–214).
2. Balthrop, J., Forrest, S., Newman, M. E. J., & Williampson, M. W. (2004). Technological networks and the spread of computer viruses. *Science*, *304*, 527–529.
3. Battista, G., Eades, P., Tamassia, R., & Tollis, I. (1999). *Graph drawing: An annotated bibliography*. Prentice-Hall, New Jersy.
4. Estévez, P. A., Figueroa, C. J., & Saito, K. (2005). Cross-entropy embedding of high-dimensional data using the neural gas model. *Neural Networks*, *18*, 727–737.
5. Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004). Information diffusion through blogspace. *Proceedings of the 13th International World Wide Web Conference* (pp. 107–117).
6. Iwata, T., Saito, K., Ueda, N., Stromsten, S., Griffiths, T. L., & Tenenbaum, J. B. (2007). Parametric embedding for class visualization. *Neural Computation*, *19*, 2536–2556.
7. Kamada, K., & Kawai, S. (1989). An algorithm for drawing general undirected graph. *Information Processing Letters*, *31*, 7–15.
8. Kempe, D., Kleinberg, J., & Tardos, E. (2003). Maximizing the spread of influence through a social network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 137–146).
9. Kimura, M., Saito, K., & Nakano, R. (2007). Extracting influential nodes for information diffusion on a social network. *Proceedings of the 22nd AAAI Conference on Artificial Intelligence* (pp. 1371–1376).
10. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*, 167–256.
11. Newman, M. E. J., Forrest, S., & Balthrop, J. (2002). Email networks and the spread of computer viruses. *Physical Review E*, *66*, 035101.
12. Newman, M. E. J. & Park, J. (2003). Why social networks are different from other types of networks. *Physical Review E*, *68*, 036122.
13. Platt, J. C. (2005). Fastmap, metricmap, and landmark mds are all nyström algorithms. *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics* (pp. 261–268).
14. Yamada, T., Saito, K., & Ueda, N. (2003). Cross-entropy directed embedding of network data. *Proceedings of the 20th International Conference on Machine Learning* (pp. 832–839).



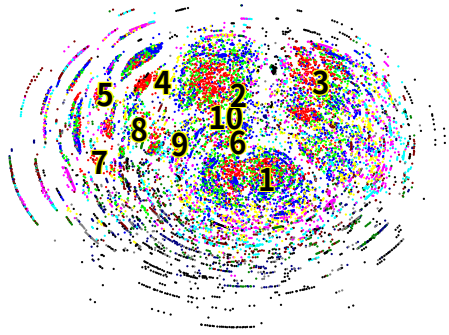
(a) Proposed method 1



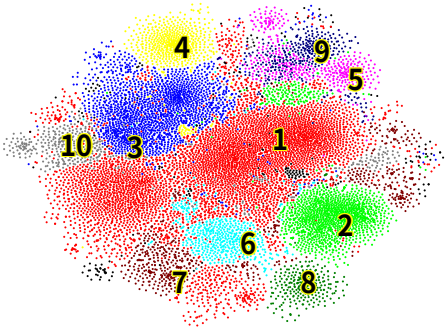
(b) Proposed method 2



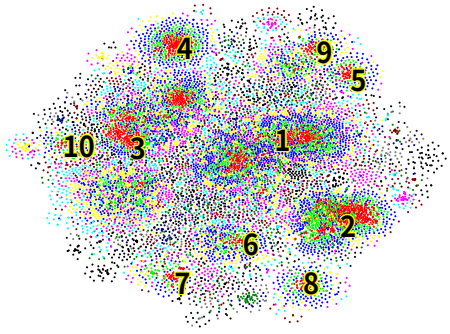
(c) Spring model method 1



(d) Spring model method 2



(e) Toplogy-based cross-entropy method 1

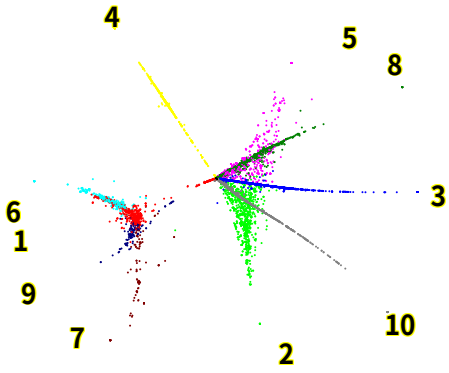


(f) Toplogy-based cross-entropy method 2

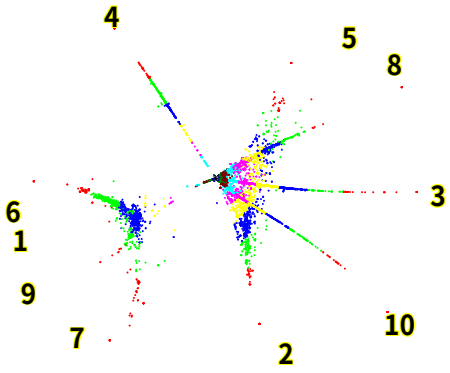
(g) Class-label assignment

(h) Color-map assignment

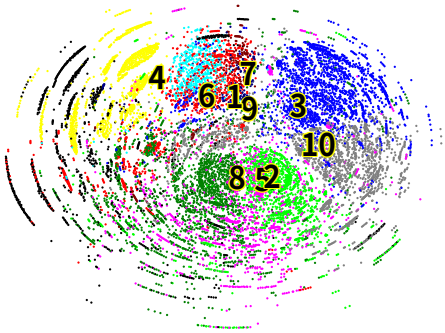
Fig. 2: Visualization of IC model for blog network



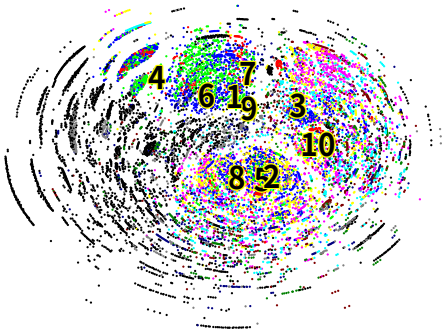
(a) Proposed method 1



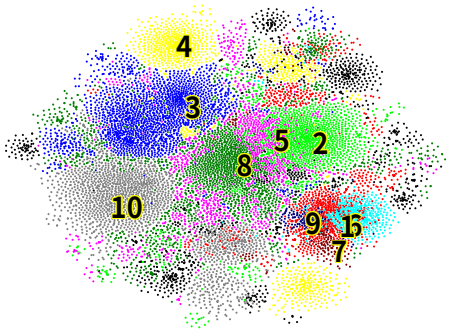
(b) Proposed method 2



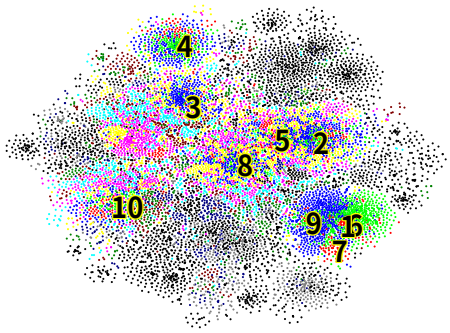
(c) Spring model method 1



(d) Spring model method 2



(e) Toplogy-based cross-entropy method 1

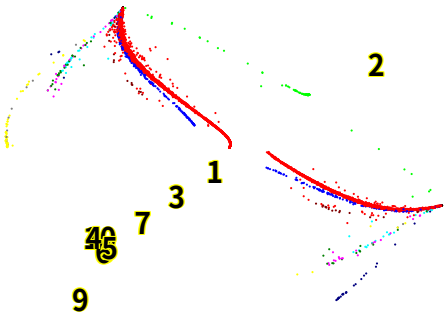


(f) Toplogy-based cross-entropy method 2

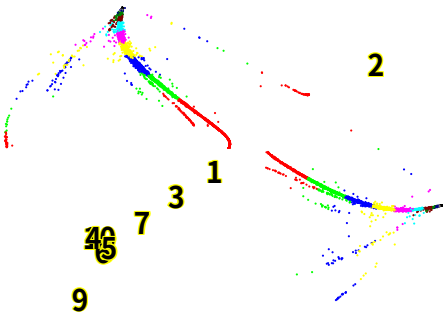
(g) Class-label assignment

(h) Color-map assignment

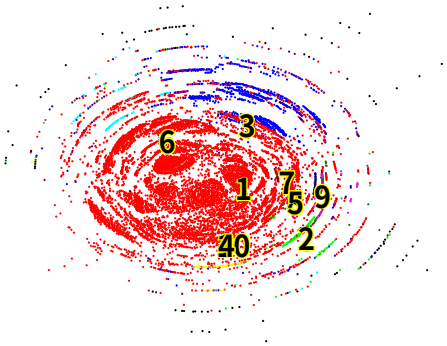
Fig. 3: Visualization of LT model for blog network



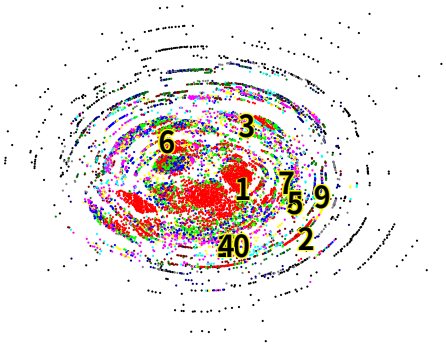
(a) Proposed method 1



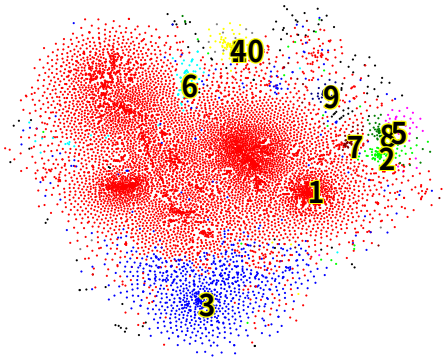
(b) Proposed method 2



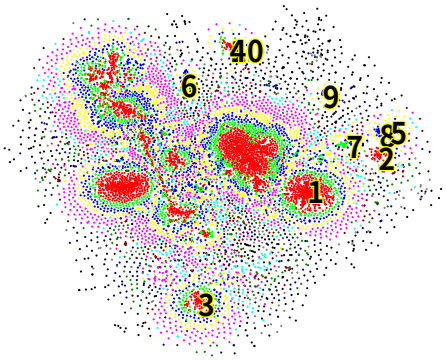
(c) Spring model method 1



(d) Spring model method 2



(e) Toplogy-based cross-entropy method 1

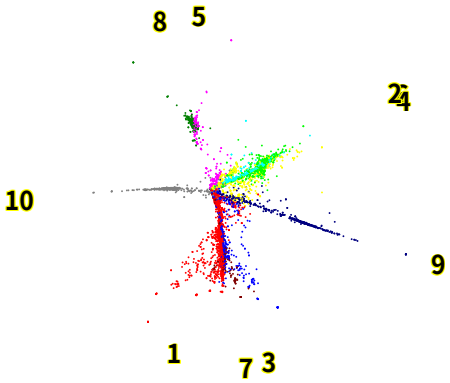


(f) Toplogy-based cross-entropy method 2

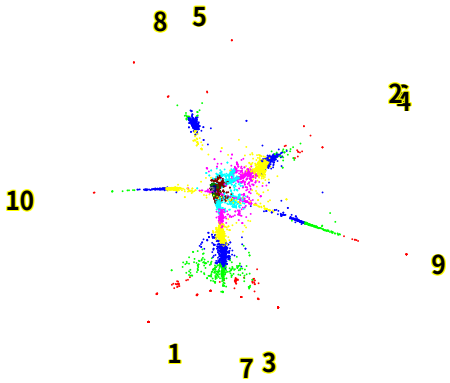
(g) Class-label assignment

(h) Color-map assignment

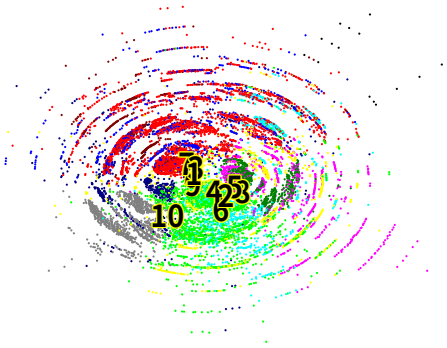
Fig. 4: Visualization of IC model for Wikipedia network



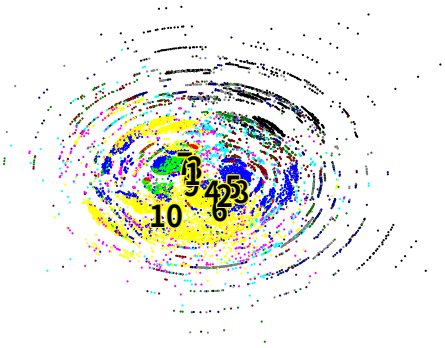
(a) Proposed method 1



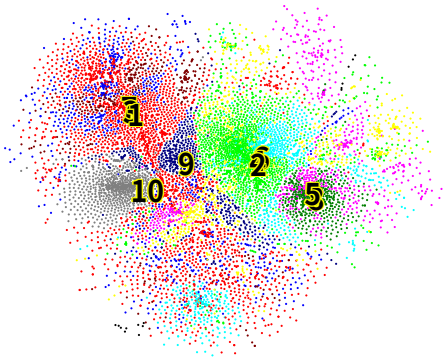
(b) Proposed method 2



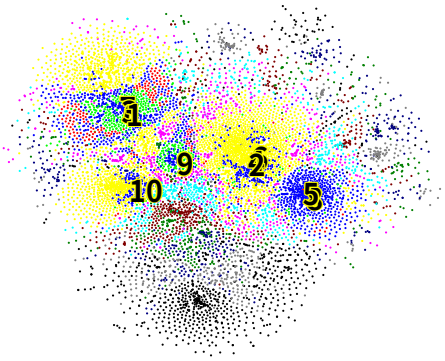
(c) Spring model method 1



(d) Spring model method 2



(e) Toplogy-based cross-entropy method 1



(f) Toplogy-based cross-entropy method 2

(g) Class-label assignment

(h) Color-map assignment

Fig. 5: Visualization of LT model for Wikipedia network

Solving the Contamination Minimization Problem on Networks for the Linear Threshold Model

Masahiro Kimura[1], Kazumi Saito[2], and Hiroshi Motoda[3]

[1] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[2] School of Administration and Informatics, University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of minimizing the spread of undesirable things, such as computer viruses and malicious rumors, by blocking a limited number of links in a network. This optimization problem called the contamination minimization problem is, not only yet another approach to the problem of preventing the spread of contamination by removing nodes in a network, but also a problem that is converse to the influence maximization problem of finding the most influential nodes in a social network for information diffusion. We adapted the method which we developed for the independent cascade model, known for a model for the spread of epidemic disease, to the contamination minimization problem under the linear threshold model, a model known for the propagation of innovation which is considerably different in nature. Using large real networks, we demonstrate experimentally that the proposed method significantly outperforms conventional link-removal methods.

1 Introduction

Networks can mediate diffusion of various things such as innovation and topics. However, undesirable things can also spread through networks. For example, computer viruses can spread through computer networks and email networks, and malicious rumors can spread through social networks among individuals. Thus, developing effective strategies for preventing the spread of undesirable things through a network is an important research issue. Previous work studied strategies for reducing the spread size by removing nodes from a network. It has been shown in particular that the strategy of removing nodes in decreasing order of out-degree can often be effective [1, 2, 3]. Here notice that removal of nodes by necessity involves removal of links. Namely, the task of removing links is more fundamental than that of removing nodes. Therefore, preventing the spread of contamination by blocking links from the underlying network is an important problem.

In contrast, finding a limted number of influential nodes that are effective for the spread of information through a social network is also an important research issue in

terms of sociology and "viral marketing" [4, 5, 6]. Widely-used fundamental probabilistic models of information diffusion through networks are the *independent cascade (IC) model* and the *linear threshold (LT) model* [7, 6]. Researchers have recently studied a combinatorial optimization problem called the *influence maximization problem* on a network under these models [7, 8]. Here, the influence maximization problem is the problem of extracting a set of k nodes to target for initial activation such that it yields the largest expected spread of information, where k is a given positive integer. Note also that the IC and LT models are fundamental models of contamination diffusion process on networks [6].

The problem we address in this paper is a problem that is converse to the influence maximization problem. The problem is to minimize the spread of contamination by blocking a limited number of links in a network. More specifically, when some undesirable thing starts with any node and diffuses through the network, we consider finding a set of k links such that the resulting network by blocking those links minimizes the expected contamination area of the undesirable thing, where k is a given positive integer. This combinatorial optimization problem is referred to as the *contamination minimization problem* [9]. For the contamination minimization problem under the IC model, Kimura, Saito and Motoda [9] presented a method for efficiently finding a good approximate solution on the basis of a naturally greedy strategy.

In this paper, we propose a method for efficiently finding a good approximate solution to the contamination minimization problem under the LT model by adapting the greedy method developed for the problem under the IC model. Note here that the IC and LT models considerably differ in quality. First, the LT model is originally a model for the propagation of innovation through the network, while the IC model can be identified with the *SIR model* [10] for the spread of epidemic disease in the network. Moreover, the LT model is viewed as a probabilistic model defined on some continous space, while the IC model is viewed as that on some finite set (i,e., a discrete space) [7, 8]. Therefore, the effectiveness of the greedy method for the problem under the LT model is not self-evident. To compare methods of solving the problem for various networks in performance, we newly introduce the *contamination reduction rate* as a performance measure. Using large real social networks, we experimentally demonstrate that the proposed method significantly outperforms link-removal heuristics that rely on the well-studied notions of betweenness and out-degree in the field of complex network theory.

2 Problem Formulation

In this paper, we address the problem of minimizing the spread of some undesirable thing in a network represented by a directed graph $G = (V, E)$. Here, V and E ($\subset V \times V$) are the sets of all the nodes and links in the network, respectively. We assume the LT model to be a mathematical model for the diffusion process of this undesirable thing in the network, and investigate the contamination minimization problem on G. We call nodes *active* if they have been contaminated by the undesirable thing.

2.1 Linear Threshold Model

We define the *linear threshold (LT) model* on graph G according to [7].

In this model, for any node $v \in V$, we specify, in advance, a *weight* $\omega_{u,v}$ (> 0) from its parent node u such that $\sum_{u \in \Gamma(v)} \omega_{u,v} \leq 1$, where $\Gamma(v)$ is the set of all the parent nodes of v, $\Gamma(v) = \{u \in V; (u, v) \in E\}$. The diffusion process from a given initial set of active nodes proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is, $\sum_{u \in \Gamma_t(v)} \omega_{u,v} \geq \theta_v$, then v will become active at time-step $t + 1$. Here, $\Gamma_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

Note that the threshold θ_v models the tendency of node v to adopt the information when its parent nodes do. Note also that the LT model is a probabilistic model associated with the uniform distribution on $[0, 1]^{|V|}$. Thus, the LT model is viewed as a probabilistic model on the continous space $[0, 1]^{|V|}$. Here, $|A|$ stands for the number of elements of a set A.

For an initial active node v, let $\sigma(v; G)$ denote the expected number of active nodes at the end of the random process of the LT model on G. We call $\sigma(v; G)$ the *influence degree* of node v in graph G.

2.2 Contamination Minimization Problem

Now, we give a mathematical definition of the contamination minimization problem on graph $G = (V, E)$.

First, we define the *contamination degree* $c(G)$ of graph G as the average of influence degrees of all the nodes in G, that is,

$$c(G) = \frac{1}{|V|} \sum_{v \in V} \sigma(v; G). \tag{1}$$

For any link $e \in E$, let $G(e)$ denote the graph $(V, E \setminus \{e\})$. We refer to $G(e)$ as the graph constructed by *blocking* e in G. Similarly, for any $D \subset E$, let $G(D)$ denote the graph $(V, E \setminus D)$. We refer to $G(D)$ as the graph constructed by *blocking* D in G. We define the *contamination minimization problem* on graph G as follows: Given a positive integer k with $k < |E|$, find a subset D^* of E with $|D^*| = k$ such that $c(G(D^*)) \leq c(G(D))$ for any $D \subset E$ with $|D| = k$.

For a large network, any straightforward method for exactly solving the contamination minimization problem suffers from combinatorial explosion. Therefore, we consider approximately solving the problem.

3 Proposed Method

We propose a method for efficiently finding a good approximate solution to the contamination minimization problem on graph $G = (V, E)$. We consider adapting the method which we developed for the IC model to the contamination minimization problem under the LT model which is considerably different in nature. Let k be the number of links to be blocked in this problem.

3.1 Geedy Algorithm

We approximately solve the contamination minimization problem on $G = (V, E)$ by the following greedy algorithm:

1. Set $D_0 \leftarrow \emptyset$.
2. Set $E_0 \leftarrow E$.
3. Set $G_0 \leftarrow G$.
4. **for** $i = 0$ to $k - 1$ **do**
5. Choose a link $e_* \in E_i$ minimizing $c(G_i(e))$, $(e \in E_i)$.
6. Set $D_{i+1} \leftarrow D_i \cup \{e_*\}$.
7. Set $E_{i+1} \leftarrow E_i \setminus \{e_*\}$.
8. Set $G_{i+1} \leftarrow (V, E_{i+1})$.
9. **end for**

Here, D_k is the set of links blocked, and represents the approximate solution obtained by this algorithm. G_k is the graph constructed by blocking D_k in graph G, that is, $G_k = G(D_k)$.

To implement this greedy algorithm, we need a method for calculating $\{c(G_i(e)); e \in E_i\}$ in Step 5 of the algorithm. However, the LT model is a stochastic process model, and it is an open question to exactly calcluate influence degrees by an efficient method [7]. Therefore, we develop a method for estimating $\{c(G_i(e)); e \in E_i\}$.

Kimura, Saito, and Nakano [8] presented the bond percolation method that efficiently estimates the influence degrees $\{\sigma(v; \tilde{G}); v \in \tilde{V}\}$ for any directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$. Thus, we can estimate $c(G_i(e))$ for each $e \in E_i$ by straightforwardly applying the bond percolation method. However, $|E_i|$ becomes very large for a large network unless i is very large. Therefore, we propose a method that can estimate $\{c(G_i(e)); e \in E_i\}$ in a more efficient manner on the basis of the bond percolation method.

3.2 Estimation Based on Bond Percolation Method

It is known that the LT model is equivalent to the following bond percolation process [7]: For any $v \in V$, we pick at most one of the incoming links to v by selecting link (u, v) with probability $\omega_{u,v}$ and selecting no link with probability $1 - \sum_{u \in \Gamma(v)} \omega_{u,v}$. Then, we declare the picked links to be "occupied" and the other links to be "unoccupied". Note here that the equivalent bond percolation process for the LT model is considerably different from that of IC model.

In the bond percolation method [8], we efficiently estimate the influence degrees $\{\sigma(v; G_i); v \in V\}$ in the following way. Let M be a sufficiently large positive integer. We perform the bond percolation process M times, and sample a set of M graphs, $\{G_i{}^m = (V, E_i{}^m); m = 1, \cdots, M\}$, constructed by the occupied links. Then, using the strongly connected decomposition of each G_i^m, we efficiently estimate the influence degrees $\{\sigma(v; G_i); v \in V\}$ as

$$\sigma(v; G_i) = \frac{1}{M} \sum_{m=1}^{M} |\mathcal{F}(v; G_i{}^m)|, \quad (v \in V), \tag{2}$$

(see [8] in detail). Here, $\mathcal{F}(v; {G_i}^m)$ denotes the set of all the nodes that are *reachable* from node v in the graph ${G_i}^m$. We say that node u is reachable from node v if there is a path from u to v along the links in the graph.

We are now in a position to give a method for efficiently estimating $\{c(G_i(e)); e \in E_i\}$ in Step 5 of the greedy algorithm. For the LT model, the weights $\{\omega_{u,v}\}$ must be specified in advance. We uniformly set the weights as follows: For any node $v \in V$, the weight $\omega_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by

$$\omega_{u,v} = \frac{1}{|\Gamma(v)| + 1}.$$

Here note that $\sum_{u \in \Gamma(v)} \omega_{u,v} < 1$ for any $v \in V$, that is, there exists a chance such that node v cannot become active even if all the parent nodes of v are active. Then, on the basis of Equations (1) and (2), and the independence of the bond percolation process, we estimate $\{c(G_i(e)); e \in E_i\}$ by

$$c\,(G_i(e)) \;=\; \frac{1}{|\mathcal{M}_i(e)|} \sum_{m \in \mathcal{M}_i(e)} \frac{1}{|V|} \sum_{v \in V} \mathcal{F}(v; {G_i}^m), \qquad (e \in E_i)$$

without applying the bond percolation method for every $e \in E_i$, where $\mathcal{M}_i(e) = \{m \in \{1, \cdots, M\}; e \notin {E_i}^m\}$. Namely, the proposed method can achieve a great deal of reduction in computational cost compared with the coventional bond percolation method.

4 Experimental Evaluation

4.1 Experimental Settings

In our experiments, we employed two sets of large real networks used in [9], the blog and Wikipedia networks, which exhibit many of the key features of social networks. These are bidirectional networks. The blog network had $12,047$ nodes and $79,920$ directed links, and the Wikipedia network had $9,481$ nodes and $245,044$ directed links.

For the proposed method, we need to specify the number M of performing the bond percolation process. In the experiments, we used $M = 10,000$ according to [8].

4.2 Comparison Methods

We compared the proposed method with two heuristics based on the well-studied notions of betweenness and out-degree in the field of complex network theory.

The *betweenness score* $b_{\tilde{G}}(e)$ of a link e in a directed graph $\tilde{G} = (\tilde{V}, \tilde{E})$ is defined as follows: $b_{\tilde{G}}(e) = \sum_{u,v \in \tilde{V}} n_{\tilde{G}}(e; u, v)/N_{\tilde{G}}(u, v)$, where $N_{\tilde{G}}(u, v)$ denotes the number of the shortest paths from node u to node v in $\tilde{G}$, and $n_{\tilde{G}}(e; u, v)$ denotes the number of those shortest paths that pass e. Here, we set $n_{\tilde{G}}(e; u, v)/N_{\tilde{G}}(u, v) = 0$ if $N_{\tilde{G}}(u, v) = 0$. Newman and Girvan [11] successfully extracted community structure in a network using the following link-removal algorithm based on betweeness:

1. Calculate betweenness scores for all links in the network.
2. Find the link with the highest score and remove it from the network.

3. Recalculate betweenness scores for all remaining links.
4. Repeat from Step 2.

In particular, the notion of betweenness can be interpreted in terms of signals traveling through a network. If signals travel from source nodes to destination nodes along the shortest paths in a network, and all nodes send signals at the same constant rate to all others, then the betweenness score of a link is a measure of the rate at which signals pass along the link. Thus, we naively expect that blocking the links with the highest betweenness score can be effective for preventing the spread of contamination in the network. Therefore, we apply the method of Newman and Girvan [11] to the contamination minimization problem. We refer to this method as the *betweenness method*.

On the other hand, previous work has shown that simply removing nodes in order of decreasing *out-degrees* works well for preventing the spread of contamination in most real networks [1, 2, 3]. Here, the out-degree of a node v means the number of outgoing links from the node v. Therefore, as a comparsion method, we consider the straighforward application of this node removal method. Namely, we employ the method of choosing nodes in decreasing order of out-degree and blocking simultaneously all the links attached to the chosen nodes. We refer to this method as the *out-degree method*. Note that the out-degree method can not be applied for all values of k to the contamination minimization problem of blocking k links.

4.3 Experimental Results

We evaluated the performance of the proposed method and compared it with that of the betweenness and out-degree methods. Clearly, the performance of a method for solving the contamination minimization problem can be evaluated in terms of the *contamination reduction rate CRR* that is defined as follows:

$$CRR \;=\; 100\,\frac{c(G) - c(G')}{c(G)}\,,$$

where G' stands for a solution graph constructed by blocking a specified number of links from the original graph G. We estimated the value of c by the bond percolation method with $M = 10,000$ (see Equations (1) and (2)), and computed the value of CRR.

Figures 1 and 2 show the contamination reduction rate CRR of the resulting network as a function of the *fraction of links blocked*, FLB, for the blog and Wikipedia networks, respectively. Here, the circles, triangles and diamonds indicate the results for the proposed, betweenness and out-degree methods, respectively. In the right figures of Figures 1 and 2, the dashed line indicates the contamination reduction rate of the network obtained by the proposed method when the number of links blocked, k, is 500. Here note that $k = 500$ means $FLB = 0.63\%$ and $FLB = 0.20\%$ in the blog and Wikipedia networks, respectively. We see that the proposed method outperformed the betweenness and out-degree methods for both the blog and the Wikipedaia networks.

These results imply that the proposed method works effectively as expected, and significantly outperforms the conventional link-removal heuristics, that is, the betweenness and out-degree methods. This shows that a significantly better link-blocking strategy for reducing the spread size of contamination can be obtained by explicitly incorporating

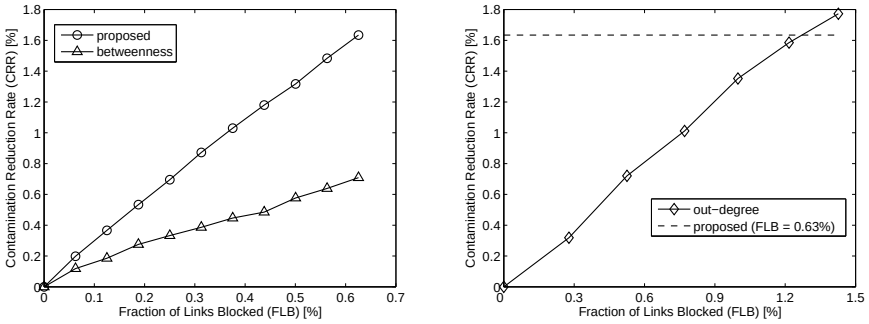


Fig. 1: Performance comparison of the proposed method with the betweenness and out-degree methods in the blog network.

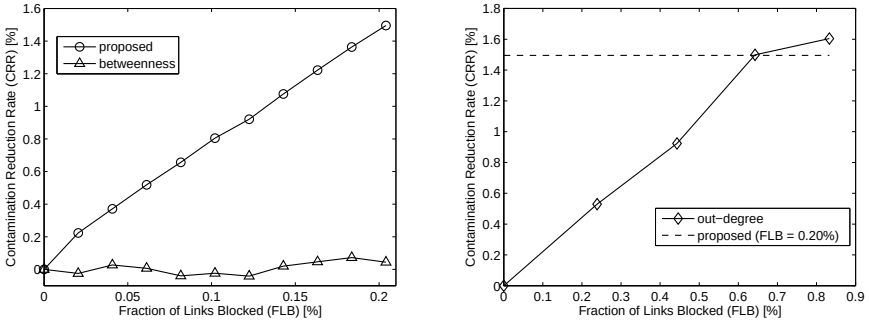


Fig. 2: Performance comparison of the proposed method with the betweenness and out-degree methods in the Wikipedia network.

the diffusion dynamics of contamination in a network, rather than relying solely on structual properties of the graph.

In the task of removing nodes from a network, the out-degree heuristic has been effective since many links can be blocked at the same time by removing nodes with high out-degrees. However, we find that in the task of blocking a limited number of links, the strategy of blocking all the links attached to nodes with high out-degrees is not necessarily effective.

5 Conclusion

In an attempt to minimize the spread of undesirable things, such as computer viruses and malicious rumors, by blocking a limited number of links in a network, we have invesitigated the contamination minimization problem for the LT model that is a fundamental diffusion model on a network. This minimization problem is, not only yet

another approach to the problem of preventing the spread of contamination by removing nodes in a network, but also a problem that is converse to the influece maximization problem of finding the most influential nodes in a social network for information diffusion. We have adapted the method which we developed for the IC model, known for a model for the spread of epidemic disease, to the contamination minimization problem under the LT model, a model known for the propagation of innovation which is considerably different in nature. Using large-scale blog and Wikipedia networks, we have experimentally demonstrated that the proposed method effectively works, and also significantly outperforms the conventional link-removal heuristics based on the betweenness and out-degree.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and Grant-in-Aid for Scientific Research (C) (No. 20500147) from Japan Society for the Promotion of Science.

References

[1] Albert, R., Jeong, H., Barabási, A.L.: Error and attack tolerance of complex networks. Nature **406** (2000) 378–382
[2] Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., Wiener, J.: Graph structure in the web. In: Proceedings of the 9th International World Wide Web Conference. (2000) 309–320
[3] Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
[4] Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2001) 57–66
[5] Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2002) 61–70
[6] Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Proceedings of the 13th International World Wide Web Conference. (2004) 107–117
[7] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003) 137–146
[8] Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence. (2007) 1371–1376
[9] Kimura, M., Saito, K., Motoda, H.: Minimizing the spread of contamination by blocking links in a network. In: Proceedings of the 23rd AAAI Conference on Artificial Intelligence. (2008) 1175–1180
[10] Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
[11] Newman, M.E.J., Girvan, M.: Finding and evaluating community structure in networks. Physical Review E **69** (2004) 026113

What Does an Information Diffusion Model Tell about Social Network Structure?

Takayasu Fushimi[1], Takashi Kawazoe[1], Kazumi Saito[1], Masahiro Kimura[2], and Hiroshi Motoda[3]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. In this paper, we attempt to answer a question "What does an information diffusion model tell about social network structure?" To this end, we propose a new scheme for empirical study to explore the behavioral characteristics of representative information diffusion models such as the IC (Independent Cascade) model and the LT (Linear Threshold) model on large networks with different community structure. To change community structure, we first construct a GR (Generalized Random) network from an originally observed network. Here GR networks are constructed just by randomly rewiring links of the original network without changing the degree of each node. Then we plot the expected number of influenced nodes based on an information diffusion model with respect to the degree of each information source node. Using large real networks, we empirically found that our proposal scheme uncovered a number of new insights. Most importantly, we show that community structure more strongly affects information diffusion processes of the IC model than those of the LT model. Moreover, by visualizing these networks, we give some evidence that our claims are reasonable.

1 Introduction

We can now obtain digital traces of human social interaction with some relating topics in a wide variety of on-line settings, like Blog (Weblog) communications, email exchanges and so on. Such social interaction can be naturally represented as a large-scale social network, where nodes (vertices) correspond to people or some social entities, and links (edges) correspond to social interaction between them. Clearly these social networks reflect complex social structures and distributed social trends. Thus, it seems worth putting some effort in attempting to find empirical regularities and develop explanatory accounts of basic functions in the social networks. Such attempts would be valuable for understanding social structures and trends, and inspiring us to lead to the discovery of new knowledge and insights underlying social interaction.

A social network can also play an important role as a medium for the spread of various information [7]. For example, innovation, hot topics and even malicious rumors can propagate through social networks among individuals, and computer viruses can diffuse through email networks. Previous work addressed the problem of tracking the propagation patterns of topics through network spaces [3, 1], and studied effective "vaccination" strategies for preventing the spread of computer viruses through networks [8, 2]. Widely-used ufundamental probabilistic models of information diffusion through networks are the *independent cascade (IC) model* and the *linear threshold (LT) model* [4, 3]. Researchers have recently investigated the problem of finding a limited number of influential nodes that are effective for the spread of information through a network under these models [4, 5]. Moreover, the influence maximization problem has recently been extended to general influence control problems such as a contamination minimization problem [6].

To deepen our understanding of social networks and accelerating study on information diffusion models, we attempt to answer a question "What does an information diffusion model tell about social network structure?" We except that such attempts derive some improved methods for solving a number of problems based on information diffusion models such as the influence maximization problem [5]. In this paper, we propose a new scheme for emperical study to explore the behavioral characteristics of representative information diffusion models such as the IC model and the LT model on large networks with different community structure. We perform extensive numerical experiments on two large real networks, one generated from a large connected trackback network of blog data, resulting in a directed graph of $12,047$ nodes and $79,920$ links, and the other, a network of people, generated from a list of people within a Japanese Wikipedia, resulting in an undirected graph of $9,481$ nodes and $245,044$ links. Through these experiments, we show that our proposed scheme could uncover a number of new insights on information diffusion processes of the IC model and the LT model.

2 Information Diffusion Models

We mathematically model the spread of information through a directed network $G = (V, E)$ under the IC or LT model, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively. We call nodes *active* if they have been influenced with the information. In these models, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set S of active nodes, we assume that the nodes in S have first become active at time-step 0, and all the other nodes are inactive at time-step 0.

2.1 Independent Cascade Model

We define the IC model. In this model, for each directed link (u, v), we specify a real value $\beta_{u,v}$ with $0 < \beta_{u,v} < 1$ in advance. Here $\beta_{u,v}$ is referred to as the *propagation probability* through link (u, v). The diffusion process proceeds from a given initial active set S in the following way. When a node u first becomes active at time-step t, it is

given a single chance to activate each currently inactive child node v, and succeeds with probability $\beta_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. If multiple parent nodes of v first become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set S, let $\varphi(S)$ denote the number of active nodes at the end of the random process for the IC model. Note that $\varphi(S)$ is a random variable. Let $\sigma(S)$ denote the expected value of $\varphi(S)$. We call $\sigma(S)$ the *influence degree* of S.

2.2 Linear Threshold Model

We define the LT model. In this model, for every node $v \in V$, we specify, in advance, a *weight* $\omega_{u,v}$ (> 0) from its parent node u such that

$$\sum_{u \in \Gamma(v)} \omega_{u,v} \leq 1,$$

where $\Gamma(v) = \{u \in V; (u,v) \in E\}$. The diffusion process from a given initial active set S proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0,1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes, u, according to weight $\omega_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is,

$$\sum_{u \in \Gamma_t(v)} \omega_{u,v} \geq \theta_v,$$

then v will become active at time-step $t+1$. Here, $\Gamma_t(v)$ stands for the set of all the parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

The LT model is also a probabilistic model associated with the uniform distribution on $[0,1]^{|V|}$. Similarly to the IC model, we define a random variable $\varphi(S)$ and its expected value $\sigma(S)$ for the LT model.

2.3 Bond Percolation Method

First, we revisit the bond percolation method [5]. Here, we consider estimating the influence degrees $\{\sigma(v;G);\ v \in V\}$ for the IC model with propagation probability p in graph $G = (V,E)$. For simplicity we assigned a uniform value p for $\beta_{u,v}$.

It is known that the IC model is equivalent to the bond percolation process that independently declares every link of G to be "occupied" with probability p [7].

It is known that the LT model is equivalent to the following bond percolation process [4]: For any $v \in V$, we pick at most one of the incoming links to v by selecting link (u,v) with probability $\omega_{u,v}$ and selecting no link with probability $1 - \sum_{u \in \Gamma(v)} \omega_{u,v}$. Then, we declare the picked links to be "occupied" and the other links to be "unoccupied". Note here that the equivalent bond percolation process for the LT model is considerably different from that of IC model.

Let M be a sufficiently large positive integer. We perform the bond percolation process M times, and sample a set of M graphs constructed by the occupied links,

$$\{G^m = (V, E^m);\ m = 1, \cdots, M\}.$$

Then, we can approximate the influence degree $\sigma(v; G)$ by

$$\sigma(v; G) \simeq \frac{1}{M} \sum_{m=1}^{M} |\mathcal{F}(v; G^m)|.$$

Here, for any directed graph $\tilde{G} = (V, \tilde{E})$, $\mathcal{F}(v; \tilde{G})$ denotes the set of all the nodes that are *reachable* from node v in the graph. We say that node u is reachable from node v if there is a path from u to v along the links in the graph. Let

$$V = \bigcup_{u \in \mathcal{U}(G^m)} \mathcal{S}(u; G^m)$$

be the strongly connected component (SCC) decomposition of graph G^m, where $\mathcal{S}(u; G^m)$ denotes the SCC of G^m that contains node u, and $\mathcal{U}(G^m)$ stands for a set of all the representative nodes for the SCCs of G^m. The bond percolation method performs the SCC decomposition of each G^m, and estimates all the influence degrees $\{\sigma(v; G);\ v \in V\}$ in G as follows:

$$\sigma(v; G) = \frac{1}{M} \sum_{m=1}^{M} |\mathcal{F}(u; G^m)|, \quad (v \in \mathcal{S}(u; G^m)), \tag{1}$$

where $u \in \mathcal{U}(G^m)$.

3 Proposed Scheme for Experimental Study

We technically describe our proposed scheme for empirical study to explore the behavioral characteristics of representative information diffusion models on large networks different community structure. In addition, we present a method for visualizing such networks in terms of community structure. Hereafter, the degree of a node v, denoted by $deg(v)$, means the number of links connecting from or to the node v.

3.1 Affection of Community Structure

As mentioned earlier, our scheme consists of two parts. Namely, to change community structure, we first construct a GR (generalized random) network from an originally observed network. Here GR networks are constructed just by randomly rewiring links of the original network without changing the degree of each node [7]. Then we plot the influence degree based on an information diffusion model with respect to the degree of each information source node.

First we describe the method for constructing a GR network. By arbitrary ordering all links in a given original network, we can prepare a link list $L_E = (e_1, \cdots, e_{|E|})$. Recall that each directed link consists of an ordered pair of *from*-part and *to*-part nodes,

i.e., $e = (u, v)$. Thus, we can produce two node lists from the list L_E, that is, the *from*-part node list L_F and the *to*-part node list L_T. Clearly the frequency of each node v appearing in L_F (or L_T) is equivalent to the out (or in) degree of the node v. Therefore, by randomly reordering the node list L_T, then concatenating it with the other node list L_F, we can produce a link list for a GR network. More specifically, let L'_T be a shuffled node list, and we denote the i-th order element of a list L by $L(i)$, then the link list of the GR network is $L'_E = ((L_F(1), L'_T(1)), \cdots, (L_F(|E|), L'_T(|E|)))$. Here note that to fairly compare the GR network with original one in terms of influence degree, we excluded some types of shuffled node lists, each of which produces a GR network with self-links of some node or multiple-links between any two nodes.

By using the bond percolation method described in the previous section, we can efficiently obtain the influence degree $\sigma(v)$ for each node v. Thus we can straightforwardly plot each pair of $deg(v)$ and $\sigma(v)$. Moreover, to examine their tendency of nodes with the same degree δ, we also plot the average influence degree $\mu(\delta)$ calculated by

$$\mu(\delta) = \frac{1}{|\{v : deg(v) = \delta\}} \sum_{\{v:deg(v)=\delta\}} \sigma(v). \tag{2}$$

Clearly we can guess that nodes with larger degrees influence many other nodes in any information diffusion models, but we consider that it is worth examining its curves in more details.

3.2 Visualization of Community Structure

In order to intuitively grasp the original and GR networks in terms of community structure, we present a visualization method that is based on the cross-entropy algorithm [11] for network embedding, and the k-core notion [10] for label assignment.

First we describe the network embedding problem. Let $\{\mathbf{x}_v : v \in V\}$ be the embedding positions of the corresponding $|V|$ nodes in an R dimensional Euclidean space. As usual, we define the Euclidean distance between $\mathbf{x}_u$ and $\mathbf{x}_w$ as follows:

$$d_{u,w} = \|\mathbf{x}_u - \mathbf{x}_w\|^2 = \sum_{r=1}^{R} (x_{u,r} - x_{w,l})^2.$$

Here we introduce a monotonic decreasing function $\rho(s) \in [0, 1]$ with respect to $s \geq 0$, where $\rho(0) = 1$ and $\rho(\infty) = 0$. Let $a_{u,w} \in \{0, 1\}$ be an adjacency information between two nodes u and w, indicating whether their exist a link between them ($a_{u,w} = 1$) or not ($a_{u,w} = 0$). Then we can introduce a cross-entropy (cost) function between $a_{u,w}$ and $\rho(d_{u,w})$ as follows:

$$\mathcal{E}_{u,w} = -a_{u,w} \ln \rho(d_{u,w}) - (1 - a_{u,w}) \ln(1 - \rho(d_{v,w})).$$

Since $\mathcal{E}_{u,w}$ is minimized when $\rho(d_{u,w}) = a_{u,w}$, this minimization with respect to $\mathbf{x}_u$ and $\mathbf{x}_w$ basically coincides with our problem setting. In this paper, we employ $\rho(s) = \exp(-s/2)$ as the monotonic decreasing function. Then the total cost function (objective function) can be defined as follows:

$$\mathcal{E} = \frac{1}{2} \sum_{u \in V} \sum_{w \in V} a_{u,w} d_{u,w} - \sum_{u \in V} \sum_{w \in V} (1 - a_{u,w}) \ln(1 - \rho(d_{u,w})). \tag{3}$$

Namely the cross-entropy algorithm minimizes the objective function defined in (3) with respect to $\{\mathbf{x}_v : v \in V\}$.

Next we explain the k-core notion. For a given node v in the network $G = (V_G, E_G)$, we denote $A_G(v)$ as a set of *adjacent nodes* of v as follows:

$$A_G(v) = \{w : \{v, w\} \in E_G\} \cup \{u : \{u, v\} \in E_G\}.$$

A subnetwork $C(k)$ of G is called *k-core* if each node in $C(k)$ has more than or equal to k adjacent nodes in $C(k)$. More specifically, we can define k-core subnetwork as follows. For a given order k, the k-core is a subnetwork $C(k) = (V_{C(k)}, E_{C(k)})$ consisting of the following node set $V_{C(k)} \subset V_G$ and link set $V_{C(k)} \subset V_G$:

$$V_{C(k)} = \{v : |A_{C(k)}(v)| \geq k\}, \quad E_{C(k)} = \{e : e \subset V_{C(k)}\}.$$

Here according to our purpose, we focus on the subnetwork of maximum size with this property as a k-core subnetwork $C(k)$.

Finally we describe the label assignment strategy. As a rough necessary condition, we assume that each community over a network includes a higher order k-core as its part. Here we consider that a candidate for such higher core order is greater than the average degree calculated by $\overline{d} = |E|/|V|$. Then we can summarize our visualization method as follows: after embedding a given network into an R (typically $R = 2$) dimensional Euclidean space by use of the cross-entropy algorithm, our visualization method plots each node position by changing the appearance of nodes belonging to its $([\overline{d}] + 1)$-core subnetwork. Here note that $[\overline{d}]$ denotes the greatest integer smaller than $\overline{d}$. By this visualization method, we can expect to roughly grasp community structure of a given network.

4 Experimental Evaluation

4.1 Network Data

In our experiments, we employed two sets of real networks used in [5], which exhibit many of the key features of social networks as shown later. We describe the details of these network data.

The first one is a trackback network of blogs. Blogs are personal on-line diaries managed by easy-to-use software packages, and have rapidly spread through the World Wide Web [3]. Bloggers (*i.e.*, blog authors) discuss various topics by using trackbacks. Thus, a piece of information can propagate from one blogger to another blogger through a trackback. We exploited the blog "Theme salon of blogs" in the site "goo" [2], where a blogger can recruit trackbacks of other bloggers by registering an interesting theme. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR Fukuchiyama Line Derailment Collision", we collected a large connected trackback network in May, 2005. The resulting network had 12,047 nodes and 79,920 directed links, which features the so-called "power-law" distributions for the out-degree and in-degree that most real large networks exhibit. We refer to this network data as the blog network.

[2] `http://blog.goo.ne.jp/usertheme/`

The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages. The undirected graph is represented by an equivalent directed graph by regarding undirected links as bidirectional ones[3]. The resulting network had 9, 481 nodes and 245, 044 directed links. We refer to this network data as the Wikipedia network.

4.2 Characteristics of Network Data

Newman and Park [9] observed that social networks represented as undirected graphs generally have the following two statistical properties that are different from non-social networks. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* C than the corresponding *configuration model* defined as the ensemble of GR networks. Here, the clustering coefficient C for an undirected network is defined by

$$C = \frac{1}{|V|} \sum_{u \in V} \frac{|\{(v \in V, w \in V) : v \neq w, w \in A_G(v)\}|}{|A_G(u)|(|A_G(u)| - 1)}.$$

Another widely-used statistical measure of network is the average length of shortest paths between any two nodes defined by

$$L = \frac{1}{|V|(|V| - 1)} \sum_{u \neq v} l(u, v).$$

where $l(u, v)$ denotes the shortest path length between nodes u and v. In terms of information diffusion processes, when L becomes smaller the probability that any information source nodes can activate the other nodes, becomes larger in general.

Table 1 shows the basic statistics of the blog and Wikipedia networks, together with their GR networks. We can see that the measured value of C for the original blog network is substantially larger than that of the GR blog network, and the measured value of L for the original blog network is somehow larger than that of the GR blog network indicating that there exisit communities. We can observe a similar tendency for the Wikipedia networks. Note that we have already confirmed for the original Wikipedia network that the degrees of adjacent nodes were positively correlated, although we derived the network from Japanese Wikipedia. Therefore, we can say that the Wikipedia network has the key features of social networks.

4.3 Experimental Settings

We describe our experimental settings of the IC and LT models. In the IC model, we assigned a uniform probability β to the propagation probability $\beta_{u,v}$ for any directed link (u, v) of a network, that is, $\beta_{u,v} = \beta$. As our β setting, we employed a reciprocal

[3] For simplicity, we call a graph with bi-directional links an undirected graph

Table 1: Basic statistics of networks.

network	$\|V\|$	$\|E\|$	C	L
original blog	12,047	79,920	0.26197	8.17456
GR blog	12,047	79,920	0.00523	4.24140
original Wikipedia	9,481	245,044	0.55182	4.69761
GR Wikipedea	9,481	245,044	0.04061	3.12848

of the average degree, i.e., $\beta = |V|/|E|$. The resulting propagation probability for the original and GR blog networks was $\beta = 0.1507$, and $\beta = 0.0387$ for the original and GR Wikipedia networks. Incidentally, these values were reasonably close to those used in former study, i.e., $\beta = 0.2$ for the blog networks and $\beta = 0.03$ for the Wikipedia networks were used in the former experiments [6].

In the LT model, we uniformly set weights as follows. For any node v of a network, the weight $\omega_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by $\omega_{u,v} = 1/|\Gamma(v)|$. This experimental setting is exactly the same as the one performed in [5].

For the proposed method, we need to specify the number M of performing the bond percolation process. In the experiments, we used $M = 10,000$ [5]. Recall that the parameter M represents the number of bond percolation processes for estimating the influence degree $\sigma(v)$ of a given initial active node v. In our preliminary experiments, we have already confirmed that the influence degree of each node for these networks with $M = 10,000$ are comparable to those with $M = 300,000$.

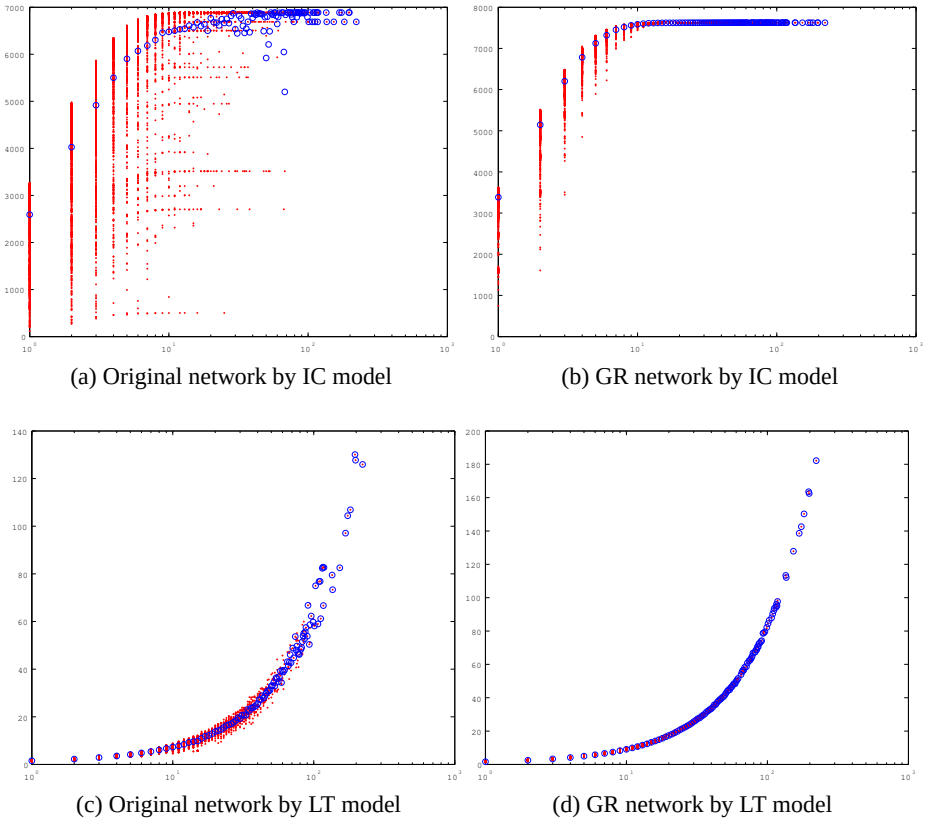
4.4 Experimental Results Using Blog Network

Figure 1a shows the influence degree based on the IC model with respect to the degree of each information source node over the original blog network, Figure 1b shows those of the IC model over the GR blog network, Figure 1c shows those of the LT model over the original Wikipedia network, and Figure 1d shows those of the LT model over the GR Wilipedia netwok. Here the red dots and blue circles respectively stand for the levels of the influence degree of individual nodes and their averages for the nodes with the same degree.

In view of the difference between the information diffusion models, we can clearly see that although nodes with larger degrees influenced many other nodes in both of the IC and LT models, their average curves exhibit opposite curvatures as shown in these results. In addition, we can observe that the influence degree of the individual nodes based on the IC model have quite large variances compared with those of the LT model.

In view of the difference between the original and GR networks, we can see that compared with the original networks, the levels of the influence degree were somewhat larger in the GR networks. We consider that this is because the averages of shortest path lengths became substantially larger than those of the GR networks, especially for the IC model. In the case of the LT model over the GR network (Figure 1d), we can observe that the influence degree was almost uniquely determined by the degree of each node. As the most remarkable point, in the case of the IC model, we can observe a number

of lateral lines composed of the individual influence degree over the original networks (Figure 1a), but these lines disappeared over the GR networks (Figure 1b).



(a) Original network by IC model (b) GR network by IC model

(c) Original network by LT model (d) GR network by LT model

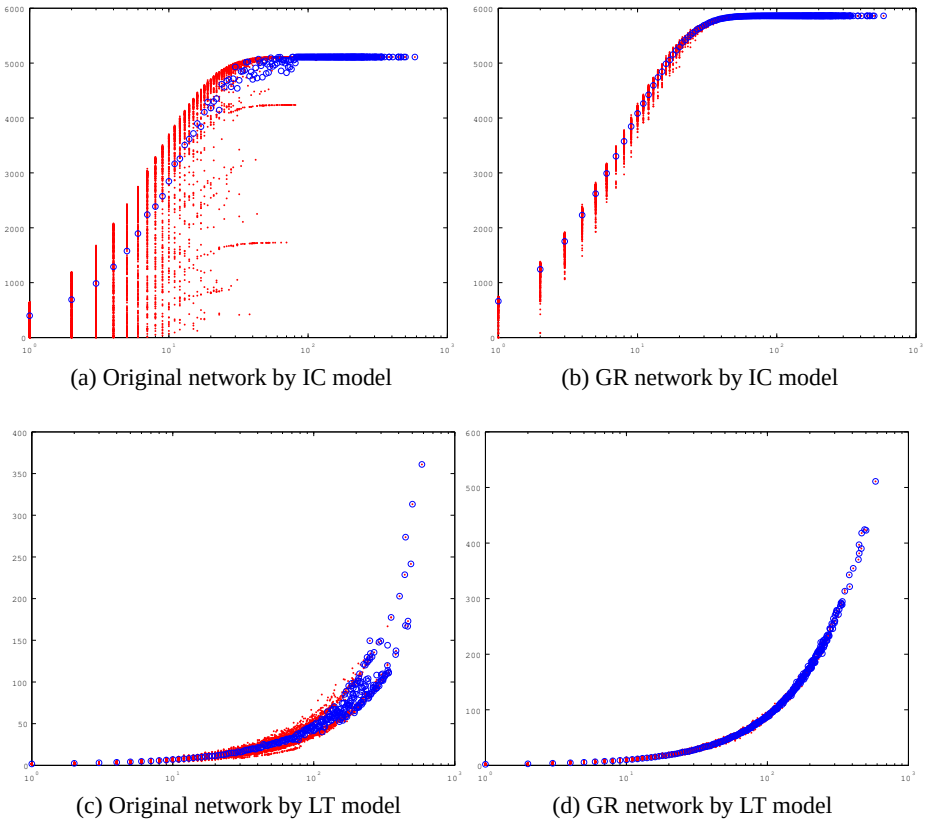
Fig. 1: Comparison of information diffusion processes using blog network

4.5 Experimental Results Using Wikipedia Network

Figure 2 shows the same experimental results using the Wikipedia networks. From these results, we can derive arguments similar to those of the blog networks. Thus we consider that our arguments were substantially strengthen by these experiments.

We summarize the main points below. 1) Nodes with larger degrees influenced many other nodes, but their average curves of the IC and LT models exhibited opposite curvatures; 2) The levels of the influence degree over the GR networks were somewhat larger than those of the original networks in both of the IC and LT models; 3) The influence degree was almost uniquely determined by the degree of each node in the case of LT model using the GR network (Figure 2d); and 4) A number of lateral lines composed

of the individual influence degree appeared in the case of IC model using the original network (Figure 2a).



(a) Original network by IC model

(b) GR network by IC model

(c) Original network by LT model

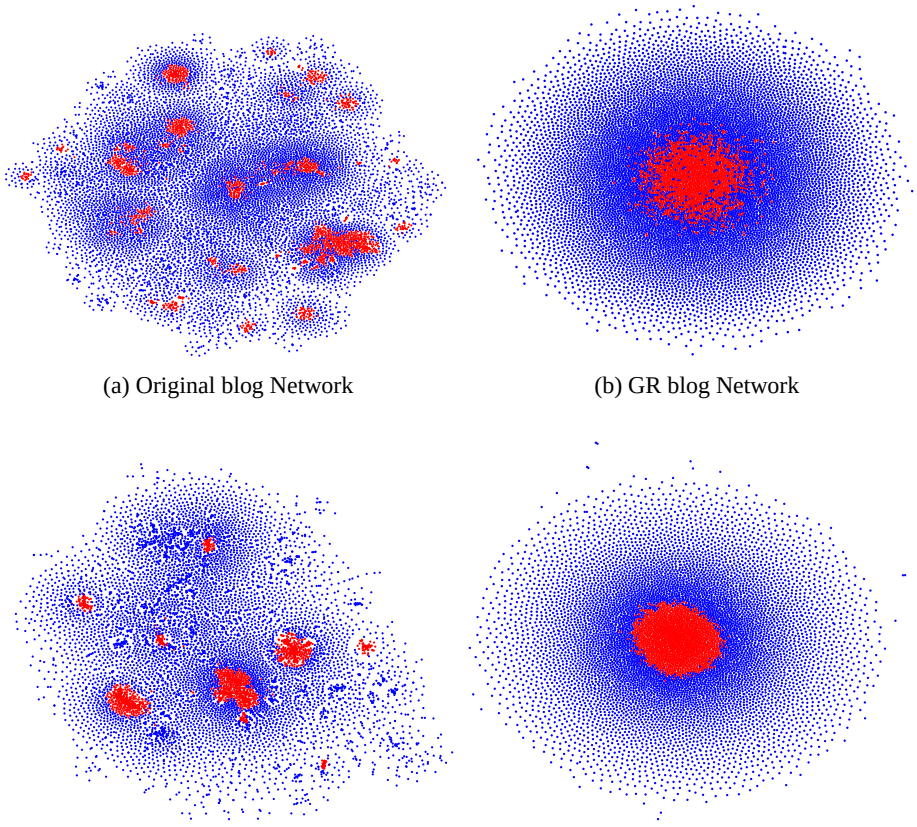
(d) GR network by LT model

Fig. 2: Comparison of information diffusion processes using wikipedia network

4.6 Community Structure Analysis

Figure 3 shows our visualization results. Here, in the case of the blog networks, since the average degree was $\overline{d} = 6.6340$, we represented the nodes belonging to the 7-core subnetwork by red points, and others by blue points. Similarly, in the case of the Wikipedia networks, since the average degree was $\overline{d} = 25.8458$, we represented the nodes belonging to the 26-core subnetwork by red points, and others by blue points. These visualization results show that the nodes of higher core order are scattered here and there in the original networks (Figures 3a and 3c), while those nodes are concentrated near the center in the GR network (Figures 3b and 3d). This clearly indicates that the transformation to GR networks changes community structure from distributed to lumped ones.

Since the main difference between the original and GR networks are their community structure, we consider that a number of lateral lines appeared in the original networks using the IC model (Figures 1a and 2a), are closely related to distributed community structure of social networks. On the other hand, we cannot observe such remarkable characteristics for the LT model (Figures 1b and 2b). In consequence, we can say that community structure more strongly affects information diffusion processes of the IC model than those of the LT model.



(a) Original blog Network (b) GR blog Network

(c) Original wikipedia Networ (d) GR wikipedia Network

Fig. 3: Visualization of Networks

5 Conclusion

In this paper, we proposed a new scheme for empirical study to explore the behavioral characteristics of representative information diffusion models such as the Independent

Cascade model and the Linear Threshold model on large networks with different community structure. The proposed scheme consists of tow parts, i.e., GR (generalized random) network construction from an originally observed network, and plotting of the influence degree of each node based on an information diffusion model. Using large real networks, we empirically found that our proposal scheme uncovers a number of new insights. Most importantly, we showed that community structure more strongly affects information diffusion processes of the IC model than those of the LT model. Our future work includes the analysis of relationships between community structure and information diffusion models by using a wide variety of social networks. We are also planing to perform further experiments by elaborating probability settings to information diffusion models.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Adar, E., & Adamic, L. (2005). Tracking information epidemics in blogspace. *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence* (pp. 207–214).
2. Balthrop, J., Forrest, S., Newman, M. E. J., & Williampson, M. W. (2004). Technological networks and the spread of computer viruses. *Science*, *304*, 527–529.
3. Gruhl, D., Guha, R., Liben-Nowell, D., & Tomkins, A. (2004). Information diffusion through blogspace. *Proceedings of the 13th International World Wide Web Conference* (pp. 107–117).
4. Maximizing the spread of influence through a social network. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 137–146).
5. Kimura, M., Saito, K., & Nakano, R. (2007). Extracting influential nodes for information diffusion on a social network. *Proceedings of the 22nd AAAI Conference on Artificial Intelligence* (pp. 1371–1376).
6. Kimura, M., Saito, K., & Motoda, H. (2008). Minimizing the spread of contamination by blocking links in a network. *Proceedings of the 23nd AAAI Conference on Artificial Intelligence* (pp. 1175–1180).
7. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, *45*, 167–256.
8. Newman, M. E. J., Forrest, S., & Balthrop, J. (2002). Email networks and the spread of computer viruses. *Physical Review E*, *66*, 035101.
9. Newman, M. E. J. & Park, J. (2003). Why social networks are different from other types of networks. *Physical Review E*, *68*, 036122.
10. S.B. Seidman, S. B. (1983). Network Structure and Minimum Degree, *Social Networks*, *5*, 269–287.
11. Yamada, T., Saito, K., & Ueda, N. (2003). Cross-entropy directed embedding of network data. *Proceedings of the 20th International Conference on Machine Learning* (pp. 832–839).

Blocking Links to Minimize Contamination Spread in a Social Network

MASAHIRO KIMURA
Ryukoku University
KAZUMI SAITO
University of Shizuoka
and
HIROSHI MOTODA
Osaka University

We address the problem of minimizing the propagation of undesirable things, such as computer viruses or malicious rumors, by blocking a limited number of links in a network, which is converse to the influence maximization problem in which the most influential nodes for information diffusion is searched in a social network. This minimization problem is more fundamental than the problem of preventing the spread of contamination by removing nodes in a network. We introduce two definitions for the contamination degree of a network, accordingly define two contamination minimization problems, and propose methods for efficiently finding good approximate solutions to these problems on the basis of a naturally greedy strategy. Using large social networks, we experimentally demonstrate that the proposed methods outperform conventional link-removal methods. We also show that unlike the case of blocking a limited number of nodes, the strategy of removing nodes with high out-degrees is not necessarily effective for these problems.

Categories and Subject Descriptors: G.2.2 [**Discrete Mathematics**]: Graph Theory—*network problems*; H.2.8 [**Database Management**]: Database Applications—*data mining*; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*sociology*

General Terms: Algorithms

Additional Key Words and Phrases: Contamination diffusion, link analysis, social networks

1. INTRODUCTION

Considerable attention has recently been devoted to investigating the structure and function of various networks including computer networks, social networks and the World Wide Web [Newman 2003]. From a functional point of view, networks can mediate diffusion of various things such as innovation and topics. However, undesirable things can also spread through networks. For example, computer viruses can spread through computer networks and email networks, and malicious rumors can spread through social networks among individuals. Thus, developing effective strategies for preventing the spread of undesirable things through a network is an important research issue. Previous work studied strategies for reducing the spread size by removing nodes from a network. It has been shown in particular that the strategy of removing nodes in decreasing order of out-degree can often be effective [Albert et al. 2000; Broder et al. 2000; Callaway et al. 2000; Newman et al. 2002]. Here notice that removal of nodes by necessity involves removal of links. Namely, the task of removing links is more fundamental than that of removing nodes, and this is the problem we address in the paper.

In contrast, finding influential nodes that are effective for the spread of information through a social network is also an important research issue in terms of sociology and "viral marketing" [Domingos and Richardson 2001; Richardson and Domingos 2002; Gruhl et al. 2004]. Recent studies include attempts to solve a combinatorial optimization problem called the *influence maximization problem* on a network under the *independent cascade (IC) model*, a widely-used fundamental probabilistic model of information diffusion[Kempe et al. 2003; Kimura et al. 2007]. Here, the influence maximization problem is the problem of extracting a set of K nodes to target for initial activation such that it yields the largest expected spread of information, where K is a given positive integer. Note also that the IC model can be identified with the so-called *susceptible/infective/recovered (SIR) model* for the spread of disease in a network [Gruhl et al. 2004].

As we see, what we address in this paper is a problem that is converse to the influence maximization problem. The problem is to minimize the spread of undesirable things by blocking a limited number of links in a network. More specifically, we consider, when some undesirable thing starts with any node and diffuses through the network under the IC model, finding a set of K links such that the resulting network obtained by blocking those links minimizes the *contamination degree* for the undesirable thing, where K is a given positive integer. We refer to this combinatorial optimization problem as a *contamination minimization problem*. We introduce two definitions for the contamination degree of a network; the *average contamination degree* and the *worst contamination degree*. According to these definitions, we formalize two contamination minimization problems; the *average contamination minimization problem* and the *worst contamination minimization problem*. The former aims to minimize the expected number of contaminated nodes (*i.e.*, the average case), and the latter aims to minimize the maximum number of contaminated nodes (*i.e.*, the worst case).

We presented in [Kimura et al. 2008] a method for efficiently finding a good approximate solution on the basis of a naturally greedy strategy for the average contamination minimization problem. In this paper, we explain the method in more detail, and propose a novel method for efficiently finding a good approximate solution on the basis of the same greedy strategy for the worst contamination minimization problem.

Furthermore, for both the average and the worst contamination minimization problems, we compare the proposed methods with a naive greedy strategy in terms of computational complexity, and show that the proposed methods can achieve a great deal of reduction in computational cost. We also present strategies for making the proposed methods computationally more efficient in practice. Finally, using large real networks that exhibit many of the key features of social networks, we experimentally demonstrate that the proposed methods outperform link-removal heuristics that rely on the well-studied notions of betweenness and out-degree in the field of complex network theory. In particular, we show that unlike the case of blocking a limited number of nodes, the strategy of removing nodes with high out-degrees is not necessarily effective for our problems.

2. INFORMATION DIFFUSION MODEL

We assume the IC model to be a mathematical model for the diffusion process of some undesirable thing on a network. We call nodes *active* if they have been contaminated by the undesirable thing.

Let $G = (V, E)$ be a directed network, where V and E ($\subset V \times V$) stand for the sets of all the nodes and (directed) links, respectively. Throughout this paper, a network means a directed network, a link means a directed link, and we also call a network a graph. According to the work of Kempe et al. [2003], we define the IC model on graph G, and recall a mathematical definition of the influence maximization problem for the IC model on graph G.

2.1 Independent Cascade Model

First, we define the IC model on graph G. In the IC model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set A of active nodes, we assume that the nodes in A have first become active at time-step 0, and all the other nodes are inactive at time-step 0. For every $e \in E$, we specify a real value p_e with $0 < p_e < 1$ in advance. Here, p_e is referred to as the *propagation probability* through link e.

The diffusion process proceeds from a given initial active set A in the following way. When a node u first becomes active at time-step t, it is given a single chance to activate each currently inactive child node w, and succeeds with probability p_e, where $e = (u, w) \in E$. Here, for a link $e' = (u', w') \in E$, nodes u' and w' are called the *parent* and *child* nodes of w' and u', respectively. If u succeeds, then w will become active at time-step $t + 1$. If multiple parent nodes of w first become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate w in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set A, let $\varphi(A; G)$ denote the number of active nodes at the end of the random process for the IC model on G. Note that $\varphi(A; G)$ is a random variable. Let $\sigma(A; G)$ denote the expected value of $\varphi(A; G)$. We call $\sigma(A; G)$ the *influence degree* of node set A on graph G. When A is in particular equal to a set of single node $\{v\}$, we simply denote $\sigma(A; G)$ by $\sigma(v; G)$, and call $\sigma(v; G)$ the influence degree of node v on graph G.

2.2 Influence Maximization Problem

Next, we recall a mathematical definition of the influence maximization problem on a network. Here, we consider maximizing the spread of desirable information through graph $G = (V, E)$. Let K be a given positive integer with $K < |V|$. Here, $|X|$ stands for the number of elements of a set X. The influence maximization problem on G for the IC model is defined as follows: Find a subset A^* of V with $|A^*| = K$ such that $\sigma(A^*; G) \geq \sigma(A; G)$ for every $A \subset V$ with $|A| = K$.

3. PROBLEM FORMULATION

We assume that some undesirable thing starts with any node in a network and diffuses through the network under the IC model. For preventing it from spreading through the network, we aim to minimize the *contamination degree* for the undesirable thing by appropriately removing a fixed number of links. Here, the contamination degree of a network is a measure of how badly the undesirable thing will contaminate the network. We give two definitions for contamination degree, and mathematically formalize two contamination minimization problems on a network.

3.1 Contamination Degree

For any graph $G = (V, E)$, we introduce two definitions for contamination degree of G.

3.1.1 *Average Contamination Degree.* We define the *average contamination degree* $c_0(G)$ of graph G as the average of influence degrees of all the nodes in G,

$$c_0(G) = \frac{1}{|V|} \sum_{v \in V} \sigma(v; G). \tag{1}$$

3.1.2 *Worst Contamination Degree.* We define the *worst contamination degree* $c_+(G)$ of graph G as the maximum of influence degrees of all the nodes in G,

$$c_+(G) = \max_{v \in V} \sigma(v; G). \tag{2}$$

3.2 Contamination Minimization Problem

According to the above definitions of contamination degree, we mathematically define the contamination minimization problems on a network, which are converse to the influence maximization problem on the network.

For any graph $G = (V, E)$, we denote by $c(G)$ both the average contamination degree $c_0(G)$ and the worst contamination degree $c_+(G)$. For any link $e \in E$, let $G(e)$ denote the graph $(V, E \setminus \{e\})$. We refer to $G(e)$ as the graph constructed by *blocking* e in G. Similarly, for any $D \subset E$, let $G(D)$ denote the graph $(V, E \setminus D)$. We refer to $G(D)$ as the graph constructed by *blocking* D in G.

We define the *contamination minimization problems* on a graph $G = (V, E)$ as follows: Given a positive integer K with $K < |E|$, find a subset D^* of E with $|D^*| = K$ such that $c(G(D^*)) \leq c(G(D))$ for any $D \subset E$ with $|D| = K$. The contamination minimization problem for $c = c_0$ is referred to as the *average contamination minimization problem*, and the contamination minimization problem for $c = c_+$ is referred to as the *worst contamination minimization problem*.

For a large network, any straightforward method for exactly solving the contamination minimization problems suffers from combinatorial explosion. Therefore, we consider approximately solving the problems.

4. PROPOSED METHOD

We propose methods for efficiently finding good approximate solutions to our contamination minimization problems. Let K be the number of links to be blocked in the problems.

4.1 Greedy Algorithm

We approximately solve the contamination minimization problems on a given graph $G_0 = (V_0, E_0)$ by the following greedy algorithm:

A1. Initialize a subset D of E_0 as $D \leftarrow \emptyset$.
A2. Initialize a graph $G = (V, E)$ as $V \leftarrow V_0$ and $E \leftarrow E_0$.
A3. Choose a link $e_* \in E$ minimizing $c(G(e))$, $(e \in E)$.
A4. Update D as $D \leftarrow D \cup \{e_*\}$.
A5. Update $G = (V, E)$ as $E \leftarrow E \setminus \{e_*\}$.
A6. Return to Step **A3** if $|D| < K$.
A7. Set $D_K \leftarrow D$.
A8. Set $G_K \leftarrow G$.

Here, D_K is the set of links blocked, and represents the approximate solution obtained by this algorithm. We refer D_K to as the *greedy solution*. G_K is the graph constructed by blocking D_K in the graph G_0, that is, $G_K = G_0(D_K)$.

To implement this greedy algorithm, we need methods for calculating

$$e_* = \arg\min_{e \in E} c(G(e)) \tag{3}$$

for a given graph $G = (V, E)$ in Step **A3** of the algorithm. The IC model is a stochastic process model, and it is an open question to exactly calculate influence degrees by an efficient method [Kempe et al. 2003]. Therefore, we must develop methods for efficiently estimating $\{c(G(e));\ e \in E\}$ for graph $G = (V, E)$.

Kimura et al. [2007] presented the bond percolation method that efficiently estimates the influence degrees $\{\sigma(v; \tilde{G});\ v \in \tilde{V}\}$ for any graph $\tilde{G} = (\tilde{V}, \tilde{E})$. Thus, in the greedy algorithm, we can estimate $c(G(e))$ for each $e \in E$ by applying the bond percolation method for the graph $G(e)$ and using Equations (1) or (2). Namely, we can simply estimate the greedy solution D_K by implementing Step **A3** of the greedy algorithm as follows:

(1) Estimate $\{c(G(e));\ e \in E\}$ by straightforwardly performing the bond percolation method $|E|$ times.
(2) Find $e_* \in E$ such that $c(G(e_*)) \leq c(G(e))$ for any $e \in E$.

We refer this strategy to as the *naive greedy strategy*. However, $|E|$ becomes very large for a large network in the greedy algorithm unless K is very large. Namely, the naive greedy strategy is not practical for large networks. Therefore, we propose more efficient methods for estimating $e_* \in E$ satisfying Equation (3) on the basis of the bond percolation method.

4.2 Bond Percolation Method

First, we revisit the bond percolation method [Kimura et al. 2007]. Here, we consider estimating the influence degrees $\{\sigma(v; G);\ v \in V\}$ for the IC model with propagation probabilities $\{p_e; e \in E\}$ on a graph $G = (V, E)$.

The *bond percolation process with occupation probabilities* $\{p_e; e \in E\}$ on graph G is the random process in which each link $e \in E$ is independently declared "occupied" with probability p_e. Note that in terms of information diffusion on a network, the

occupied links represent the links through which the information propagates, and the unoccupied links represent the links through which the information does not propagate. For a positive integer M, we perform the bond percolation process M times, and sample a set of M graphs constructed by the occupied links,

$$\{G^m = (V, E^m);\ m = 1, \cdots, M\}.$$

For any $v \in V$, we define $s(v; G, M)$ by

$$s(v; G, M) = \frac{1}{M} \sum_{m=1}^{M} |F(v; G^m)|. \tag{4}$$

Here, for any graph $\tilde{G} = (\tilde{V}, \tilde{E})$ and any node $v \in \tilde{V}$, $F(v; \tilde{G})$ stands for the set of all the nodes that are *reachable* from node v on graph $\tilde{G}$. We say that node u is reachable from node v on graph $\tilde{G}$ if there is a path from u to v along the links on graph $\tilde{G}$.

It is known [Newman 2003] that the IC model with propagation probabilities $\{p_e; e \in E\}$ on graph G can be exactly mapped onto the bond percolation process with occupation probabilities $\{p_e; e \in E\}$ on graph G, and the influence degree $\sigma(v; G)$ of node $v \in V$ can well be approximated by $s(v; G, M)$,

$$\sigma(v; G) \simeq s(v; G, M), \quad (v \in V), \tag{5}$$

if M is sufficiently large. We decompose each graph G^m into the strongly connected components (SCCs) as follows:

$$V = \bigcup_{j=1}^{J^m} SCC(u_j^m; G^m), \tag{6}$$

where J^m is the number of the strongly connected components of graph G^m, each u_j^m is an element of V, and $SCC(u_j^m; G^m)$ denotes the SCC of graph G^m that contains node u_j^m. Note that

$$|F(v; G^m)| = \left|F(u_j^m; G^m)\right|, \quad \text{if } v \in SCC(u_j^m; G^m). \tag{7}$$

Thus, by calculating $\{|F(u_j^m; G^m)|; j = 1, \cdots, J^m\}$ in advance and using Equation (7), we efficiently calculate $|F(v; G^m)|$ for all $v \in V$. Once we have $\{|F(v; G^m)|; v \in V, m = 1, \cdots, M\}$, we can calculate $s(v; G, M)$ for all $v \in V$ from Equation (4).

Namely, the bond percolation method estimates all the influence degrees $\{\sigma(v; G); v \in V\}$ on graph G as follows: It first specifies the value of integer M, calculates $s(v; G, M)$ for all $v \in V$ by performing the above procedure, and estimates $\sigma(v; G)$ for all $v \in V$ by using Equation (5).

4.3 Estimation Method

Now, we give methods for efficiently estimating $e_* \in E$ satisfying Equation (3) for a given graph $G = (V, E)$ to implement Step **A3** of the greedy algorithm for the average and the worst contamination minimization problems.

First, we perform the bond percolation process M times on graph $G = (V, E)$, and sample a set of M graphs constructed by the occupied links,

$$\{G^m = (V, E^m);\ m = 1, \cdots, M\},$$

where M is a given positive integer. Next, we calculate

$$\mathcal{B}_M(e) = \{m \in \{1, \cdots, M\};\ e \notin E^m\}, \quad (e \in E). \tag{8}$$

Note that $\mathcal{B}_M(e)$ represents the subset of the M trials for the bond percolation process on graph G such that e is not an occupied link.

Here, we consider performing the bond percolation process $|\mathcal{B}_M(e)|$ times on the graph $G(e) = (V, E \setminus \{e\})$ for any $e \in E$, and sampling a set of $|\mathcal{B}_M(e)|$ graphs constructed by the occupied links,

$$\{G(e)^m;\ m = 1, \cdots, |\mathcal{B}_M(e)|\}.$$

We assume that M is large enough so that $|\mathcal{B}_M(e)|$ is also sufficiently large. Then, by Equation (5), we have

$$\sigma(v; G(e)) \simeq s\left(v; G(e), |\mathcal{B}_M(e)|\right), \quad (v \in V). \tag{9}$$

Note from Equation (4) that

$$s\left(v; G(e), |\mathcal{B}_M(e)|\right) = \frac{1}{|\mathcal{B}_M(e)|} \sum_{m=1}^{|\mathcal{B}_M(e)|} |F(v; G(e)^m)|, \quad (v \in V). \tag{10}$$

In order to efficiently estimate $\{c(G(e));\ e \in E\}$ without applying the bond percolation method on the graph $G(e)$ for every $e \in E$, we alternatively calculate

$$\overline{s}_M(v, e) = \frac{1}{|\mathcal{B}_M(e)|} \sum_{m \in \mathcal{B}_M(e)} |F(v; G^m)|, \quad (v \in V,\ e \in E), \tag{11}$$

for the graph G on the basis of the bond percolation method. Since each link of graph G is independently declared "occupied" in the bond percolation process, we can obtain the following theorem from Equations (8), (9), (10) and (11).

THEOREM 4.1. *Let $G = (V, E)$ be a graph. For every $v \in V$ and $e \in E$, we have*

$$\overline{s}_M(v, e) \to \sigma(v; G(e))$$

as $M \to \infty$.

From Theorem 4.1, we can apply the approximation

$$\sigma(v; G(e)) \simeq \overline{s}_M(v, e), \quad (v \in V,\ e \in E), \tag{12}$$

for a sufficiently large M. Therefore, by Equations (1) and (2), we propose estimating $e_* \in E$ satisfying Equation (3) for a given graph $G = (V, E)$ as follows:

$$e_* = \arg\min_{e \in E} \left(\frac{1}{|V|} \sum_{v \in V} \overline{s}_M(v, e) \right) \tag{13}$$

for the average contamination minimization problem (*i.e.*, $c = c_0$), and

$$e_* = \arg\min_{e \in E} \left(\max_{v \in V} \overline{s}_M(v, e) \right) \tag{14}$$

for the worst contamination minimization problem (*i.e.*, $c = c_+$). Notice that for the proposed method, the value of M is specified in advance.

4.4 Computational Complexity and Implementational Strategy

For both the average and the worst contamination minimization problems, we compare the proposed methods with the naive greedy strategy in terms of computational complexity. We focus on the computational complexity of estimating $e_* \in E$ satisfying Equation (3) for a given graph $G = (V, E)$.

Let Q be the expected computational complexity for calculating the values of $\{s(v; G, 1);\ v \in V\}$ on graph $G = (V, E)$ on the basis of the bond percolation method (see, Equation (4)). Then, the expected computational complexity of the proposed method for calculating $\{\overline{s}_M(v, e);\ v \in V,\ e \in E\}$ amounts to MQ, since the values of $\{|F(v; G^m)|;\ v \in V,\ m = 1, \cdots, M\}$ are calculated on the basis of the bond percolation method (see, Equations (4) and (11)). Note that for any $e \in E$, calculating $\{\overline{s}_M(v, e);\ v \in V\}$ for the proposed methods corresponds to estimating $c(G(e))$ through $|\mathcal{B}_M(e)|$ trials of the bond percolation process on graph $G(e)$ (see, Equation (11)). For the naive greedy strategy, we consider estimating $c(G(e))$ through $|\mathcal{B}_M(e)|$ trials of the bond percolation process on graph $G(e)$ (see, Equations (9) and (10)). Then, in order to estimate the values of $\{c(G(e);\ e \in E\}$, the naive greedy strategy requires the computational complexity of $Q \sum_{e \in E} |\mathcal{B}_M(e)|$. Here we assumed that the computational complexities of $s(v; G, 1)$ and $s(v; G(e), 1)$ are the same because $|E|$ is sufficiently large in general. By noting that the expected value of $|\mathcal{B}_M(e)|$ is $(1 - p_e)M$, the expected computational complexity of the naive greedy strategy for estimating $\{c(G(e);\ e \in E\}$ becomes $MQ \sum_{e \in E}(1 - p_e)$. Thus, we can see that the proposed methods are $\sum_{e \in E}(1 - p_e)$ times faster than the naive greedy strategy on average. For instance, when the number of links is $100,000$ and each propagation probability p_e for the IC model is a uniform probability $p = 0.2$, the value of $\sum_{e \in E}(1 - p_e)$ is $80,000$. Namely, the proposed methods can achieve a great deal of reduction in computational cost, compared with the naive greedy strategy.

Furthermore, the following strategies can be used to efficiently find $e_* \in E$ satisfying Equations (13) or (14) for a given graph $G = (V, E)$ in actual practice.

First, as for the worst contamination minimization problem, we apply the idea of lazy evaluations for marginal increments of a submodular function by Leskovec et al. [2007]. More specifically, we efficiently calculate Equation (14) by appropriately pruning the evaluations for $\{\overline{s}_M(v, e);\ v \in V,\ e \in E\}$. By Equations (4) and (11), we have

$$M\, s(v; G, M) \ = \ |\mathcal{B}_M(e)|\ \overline{s}_M(v, e) \ + \sum_{m \in \{1, \cdots, M\} \setminus \mathcal{B}_M(e)} |F(v; G^m)|$$

for any $v \in V$ and $e \in E$. Thus, we can derive the following upper bound with respect to $\overline{s}_M(v, e)$:

$$\frac{M}{|\mathcal{B}_M(e)|} s(v; G, M) \geq \overline{s}_M(v, e), \quad (v \in V, e \in E). \tag{15}$$

We arbitrarily fix a link $e \in E$. Then, we first sort all the nodes $\{v \in V\}$ of graph G by the value $Ms(v; G, M)/|\mathcal{B}_M(e)|$ in descending order as follows: $\langle v_i;\ i = 1, \cdots, |V|\rangle$. We next calculate the value of $\overline{s}_M(v, e)$ in this order, until the current maximum value $\overline{s}_M(v_i^*, e)$ exceeds the value $Ms(v_{i+1}; G, M)/|\mathcal{B}_M(e)|$ for the head v_{i+1} of the remaining nodes. By Equation (15), this pruning guar-

antees that the current maximum value attains the maximum without necessarily evaluating $\overline{s}_M(v,e)$ for all $v \in V$. In our experiments, the computational efficiency was greatly improved by using this strategy, just as reported in [Leskovec et al. 2007].

Next, as for the average contamination minimization problem, we efficiently calculate Equation (13) without evaluating the value of $\overline{s}_M(v,e)$ for every pair of node v and link e. Our strategy is to exploit the relation

$$\frac{1}{|V|}\sum_{v\in V}\overline{s}_M(v,e) \;=\; \frac{1}{|\mathcal{B}_M(e)|}\sum_{m\in\mathcal{B}_M(e)}\frac{1}{|V|}\sum_{v\in V}|F(v;G^m)|, \quad (v\in V, e\in E), \quad (16)$$

(see, Equation (11)). More specifically, we evaluate $\sum_{v\in V}|F(v;G^m)/|V|$ for each m on the basis of the bond percolation method in advance (see, Equations (6) and (7)), and then calculate Equation (13) by evaluating $\sum_{v\in V}\overline{s}_M(v,e)$ for every $e \in E$ using Equation (16).

5. EXPERIMENTAL EVALUATION

Using two large real networks that exhibit many of the key features of social networks, we experimentally evaluated the performance of the proposed method.

5.1 Network Data

First, we employed a trackback network of blogs because a piece of information can propagate from one blog author to another blog author through a trackback. Since bloggers (*i.e.*, blog authors) discuss various topics and establish mutual communications by putting trackbacks on each other's blogs, we regarded a link created by a trackback as a bidirectional link. By tracing up to ten steps back in the trackbacks from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo" [1], we collected a large connected trackback network in May, 2005. The resulting network was a directed graph of $12,047$ nodes and $79,920$ links, which features the so-called "power-law" degree distribution that most large real networks exhibit (see, Figure 1). Here, the degree distribution is the distribution of the number of undirected links for every node. We refer to this network data as the blog network.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed graph regarding those undirected links as bidirectional ones. We refer to this network data as the Wikipedia network. Here, the total numbers of nodes and directed links were $9,481$ and $245,044$, respectively. The network also showed the power-law degree distribution (see, Figure 2).

Newman and Park [2003] observed that social networks represented as undirected graphs generally have the following two statistical properties that are different from non-social networks. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient*

[1] `http://blog.goo.ne.jp/usertheme/`

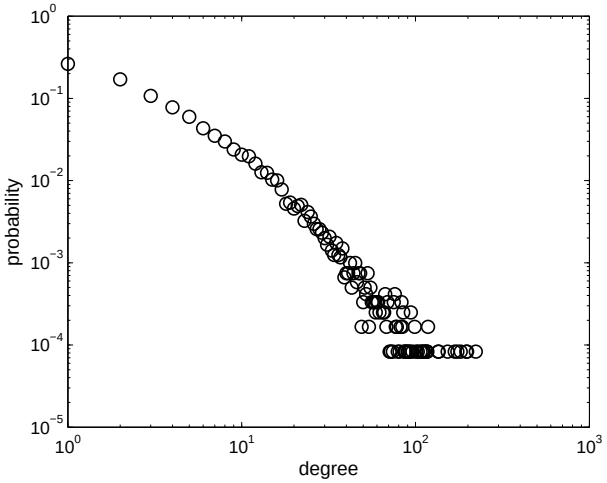


Fig. 1. The degree distribution for the blog network.

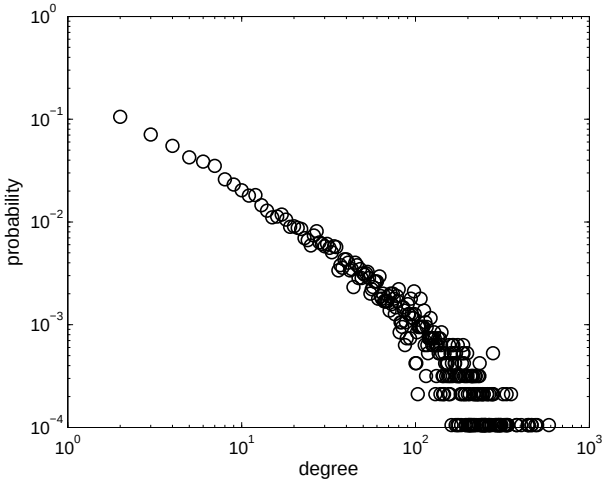


Fig. 2. The degree distribution for the Wikipedia network.

CC than the corresponding *configuration models* (*i.e.*, random network models). Here, the clustering coefficient CC for an undirected graph is defined by

$$CC = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of nodes}},$$

where a "triangle" means a set of three nodes each of which is connected to each other, and a "connected triple" means a node connected directly to unordered other pair nodes. For the undirected graph of the Wikipedia network, the value of CC of the corresponding configuration model was 0.046, while the actual measured value

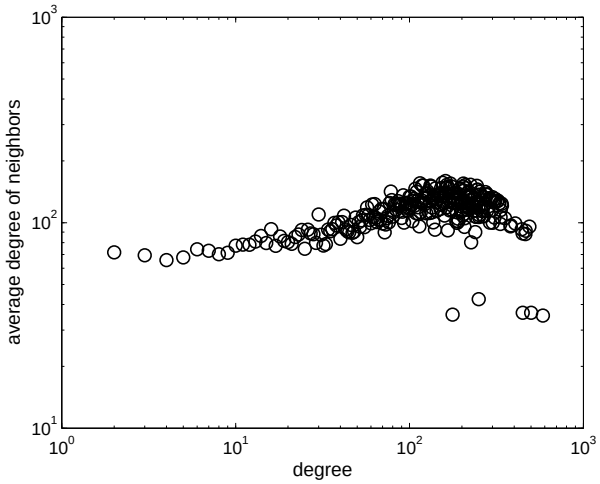


Fig. 3. The degree correlation for the Wikipedia network.

of CC was 0.39. Namely, the undirected graph of the Wikipedia network had a much higher value of the clustering coefficient than the corresponding configuration model. Moreover, we can see from Figure 3 that the Wikipedia network had weakly positive degree correlation. Therefore, we believe that the Wikipedia network is a typical example of a large real social network represented by an undirected graph, and can be used as the network data to evaluate the performance of the proposed method.

5.2 Experimental Settings

For the bond percolation method, we need to specify the number M of performing the bond percolation process. It is reported [Kimura et al. 2007] that setting the value of M at several thousand is good enough for estimating influence degrees for the blog and Wikipedia networks. The following is the basis of assessing the value of M in the experiments in this paper. We estimated the average and the worst contamination degrees for the two networks with $M = 8,000$ and $M = 300,000$, where we assigned a uniform probability p to each propagation probability p_e for the IC model (how the value of p is determined for each network is described in detail in the next paragraph). The difference in the estimated average contamination degree for $M = 8,000$ and $M = 300,000$ was about 0.01% for the blog network and 0.02% for the Wikipedia network. Also, the corresponding difference in the estimated worst contamination degree was about 0.02% for the blog network and 0.01% for the Wikipedia network. Thus, we concluded that the estimated contamination degrees for these networks with $M = 8,000$ are comparable to those with $M = 300,000$. By considering the assigned values of the propagation probabilities, we decided to use $M = 10,000$ through the experiments.

Because we assigned a uniform probability p to the propagation probability p_e for any directed link e of a network, the IC model had a single parameter p, and we determined the typical value of p for each of the blog and Wikipedia networks,

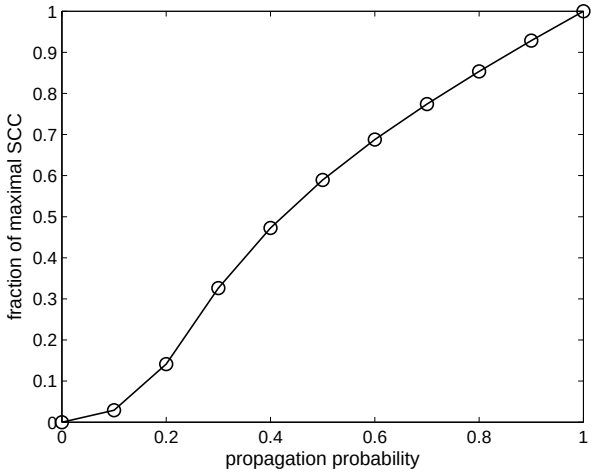


Fig. 4. Fragmentation of the blog network for the IC model. The fraction H of the maximal SCC as a function of the propagation probability p.

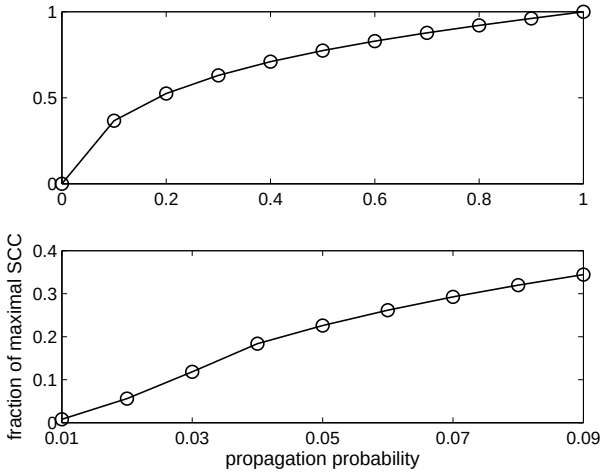


Fig. 5. Fragmentation of the Wikipedia network for the IC model. The fraction H of the maximal SCC as a function of the propagation probability p. The upper and lower frames show the network fragmentation curves for the whole range of p and the range of $0.01 \leq p \leq 0.09$, respectively.

and used them in the experiments. Let us consider the bond percolation process corresponding to the IC model with propagation probability p on a graph $G = (V, E)$. Let H be the expected fraction of the maximal SCC in the network constructed by occupied links. H is a function of p, and as the value of p decreases, the value of H decreases. In other words, as the value of p decreases, the original graph G gradually fragments into small clusters under the corresponding bond per-

colation process. Figures 4 and 5 show the network fragmentation curves for the blog and Wikipedia networks, respectively. Note that $H \to 1$ as $p \to 1$ since the blog and Wikipedia networks are strongly connected. Here, given the value of p, we estimated H as follows (see, Equation (6)):

$$H = \frac{1}{M|V|} \sum_{m=1}^{M} \max_{1 \le j \le J^m} |SCC(u_j^m; G^m)|,$$

where $M = 10,000$. We focus on the point p_* at which the average rate dH/dp of change of H attains the maximum, and regard it as the typical value of p for the network. Note that p_* is a critical point of dH/dp, and defines one of the features intrinsic to the network. From Figures 4 and 5, we estimated p_* to be $p_* = 0.2$ for the blog network and $p_* = 0.03$ for the Wikipedia network.

5.3 Comparison Methods

We compared the proposed method with three other heuristic methods. Two of them are based on the well-studied notions of betweenness and out-degree in the field of complex network theory and the other one is the crude baseline of blocking links randomly. We refer to these methods as *betweenness method, out-degree method* and *random method*, respectively.

5.3.1 *Betweenness Method.* The *betweenness score* $b_G(e)$ of a link e in a graph $G = (V, E)$ is defined as follows:

$$b_G(e) = \sum_{u,v \in V} \frac{n_G(e; u, v)}{N_G(u, v)},$$

where $N_G(u, v)$ denotes the number of the shortest paths from node u to node v on graph G, and $n_G(e; u, v)$ denotes the number of those paths that pass e. Here, we set $n_G(e; u, v)/N_G(u, v) = 0$ if $N_G(u, v) = 0$. Newman and Girvan [2004] successfully extracted community structure in a network using the following link-removal algorithm based on betweenness:

B1. Calculate betweenness scores for all links in the network.

B2. Find the link with the highest score and remove it from the network.

B3. Recalculate betweenness scores for all remaining links.

B4. Repeat from Step **B2**.

In particular, the notion of betweenness can be interpreted in terms of signals traveling through a network. If signals travel from source nodes to destination nodes along the shortest paths in a network, and all nodes send signals at the same constant rate to all others, then the betweenness score of a link is a measure of the rate at which signals pass along the link. Thus, we naively expect that blocking the links with the highest betweenness score can be effective for preventing the spread of contamination in the network. Therefore, we apply the method of Newman and Girvan [2004] to the contamination minimization problems.

5.3.2 *Out-degree Methods.* Previous work has shown that simply removing nodes in order of decreasing *out-degrees* works well for preventing the spread of contamination in most real networks [Albert et al. 2000; Broder et al. 2000; Callaway

et al. 2000; Newman et al. 2002]. Here, the out-degree $d(v)$ of a node v means the number of outgoing links from the node v. Therefore, as a comparison method, we consider the straightforward application of this node removal method. Namely, we employ the method of choosing nodes in decreasing order of out-degree and blocking simultaneously all the links attached to the chosen nodes. We refer to this method as the *node out-degree method*. Note that the node out-degree method cannot be applied for all values of positive integer K ($\leq |E|$) to the contamination minimization problems of blocking K links.

We also consider the method of blocking links between nodes with high out-degrees as an alternative comparison method. We define the link out-degree $\overline{d}(e)$ of a link $e = (u, v)$ from node u to node v by

$$\overline{d}(e) \;=\; d(u)\, d(v),$$

and recursively block links in decreasing order of link out-degree. We refer to this method as the *link out-degree method*.

5.4 Experimental Results

We evaluated the performance of the proposed method and compared it with that of the betweenness, the node out-degree, the link out-degree and the random methods. Clearly, the performance can be evaluated by the average contamination degree c_0 and the worst contamination degree c_+. We estimated these values by using the bond percolation method with $M = 10,000$, that is,

$$\begin{aligned} c_0(G_K) \;&=\; \frac{1}{|V|} \sum_{v \in V} s(v; G_K, M), \\ c_+(G_K) \;&=\; \max_{v \in V} s(v; G_K, M), \end{aligned}$$

(see, Equation (4)), where $M = 10,000$. Note that this evaluation is done separately from the approximation used to search for the link to be deleted, *i.e.*, Equation (11).

5.4.1 *Average Contamination Minimization Problem.* Figures 6 and 7 show the average contamination degree c_0 as a function of the number K of links blocked for the blog network and Figures 8 and 9 show the corresponding results for the Wikipedia network. In these figures the circles, squares, diamonds, triangles and crosses indicate the results for the proposed, the betweenness, the node out-degree, the link out-degree and the random methods, respectively. For each dataset, there are two figures, one comparing the proposed method with the betweenness method and the other comparing the proposed method at a fixed value of $K = 500$ with the node out-degree, the link out-degree and the random methods.

First, note that the average contamination degree c_0 at $K = 0$ is 976 for the blog network and 403 for the Wikipedia network, which is 8.2% and 4.2% respectively. The average contamination degree as defined by Equation (1) is less than 10%. The fact that this value for Wikipedia network is about half of that of the blog network is explained by the smaller value of p for the Wikipedia network with the difference in network sizes considered. As expected the proposed method performs the best and the betweenness method follows. The other three methods are much worse than these two in the networks used.

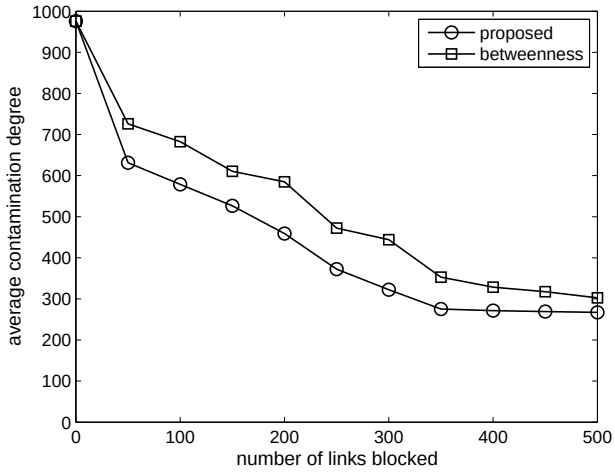


Fig. 6. Performance comparison between the proposed and the betweenness methods in the blog network for the average contamination minimization problem.

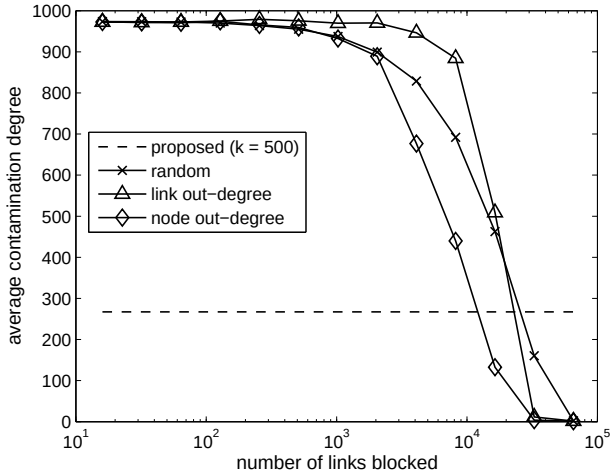


Fig. 7. Performance comparison of the proposed method at $K = 500$ with the node out-degree, the link out-degree and the random methods in the blog network for the average contamination minimization problem.

The number of links blocked: $K = 500$ corresponds to 0.63% of the total links for the blog network and 0.2% for the Wikipedia network. Inversely, 0.2% of the total links corresponds to 163 links for the blog network. The average contamination degree at 0.2% link block, *i.e.*, $K = 163$ for the blog network and $K = 500$ for the Wikipedia network is 495 and 243 for the proposed method, which is equivalent to 49% and 40% reduction in the degree, respectively, and 607 and 306 for the

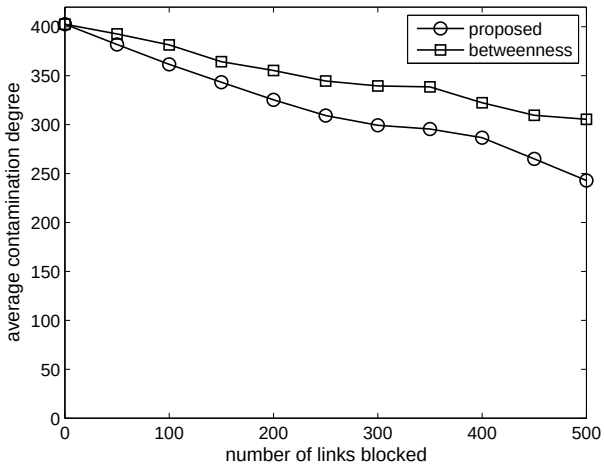


Fig. 8. Performance comparison between the proposed and the betweenness methods in the Wikipedia network for the average contamination minimization problem.

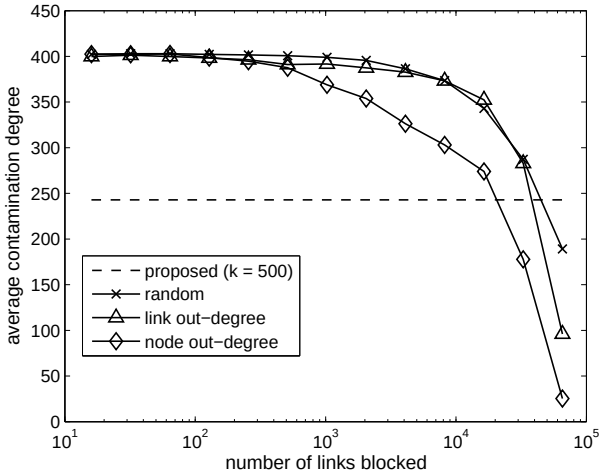


Fig. 9. Performance comparison of the proposed method at $K = 500$ with the node out-degree, the link out-degree and the random methods in the Wikipedia network for the average contamination minimization problem.

betweenness method, which is equivalent to 38% and 24% reduction in the degree, respectively. The difference between the two methods is 11% for the blog network and 16% for the Wikipedia network, respectively. The average contamination degree at 0.63% link block for the blog network, *i.e.*, $K = 500$ is 267 for the proposed method and 303 for the betweenness method, which is equivalent to 73% and 69% reduction in the degree, respectively, and the difference between the two methods

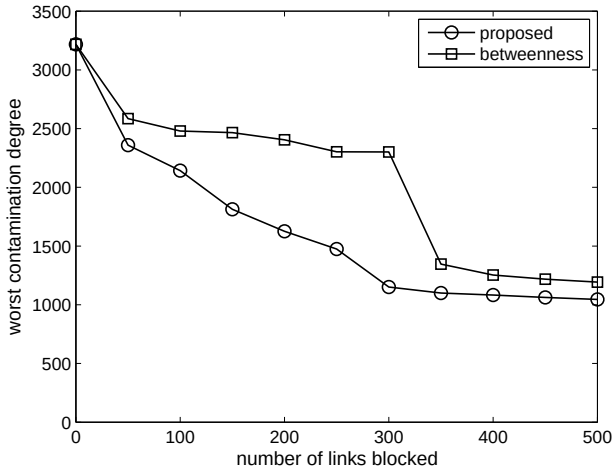


Fig. 10. Performance comparison between the proposed and betweenness methods in the blog network for the worst contamination minimization problem.

is 4%.

Differently from the above, the proposed method as well as the betweenness method outperform by far the other three methods (the node out-degree, the link out-degree and the random) for both the blog and the Wikipedia networks. Blocking 500 links by the proposed methods is equivalent to blocking more than 10,000 links for the blog network and 20,000 links for the Wikipedia network by the other three methods, meaning that the proposed method is 20 to 40 times more effective.

5.4.2 *Worst Contamination Minimization Problem.* Figures 10 and 11 show the worst contamination degree c_+ as a function of the number K of links blocked for the blog network, and Figures 12 and 13 show the corresponding results for the Wikipedia network. The meaning of the symbols in captions and the layout of the figures are the same as before.

First note that the worst contamination degree c_+ at $K = 0$ is 3218 for the blog network and 1929 for the Wikipedia network, which is 27% and 20% respectively. They are about 3 and 5 times larger than the average contamination degrees. The difference of the values between the two networks is consistent with the average contamination case. The overall performance difference among the four methods is also consistent with the average contamination case.

The worst contamination degree at 0.2% link block, *i.e.*, $K = 163$ for the blog network and $K = 500$ for the Wikipedia is 1763 and 1177 for the proposed method, which is equivalent to 45% and 39% reduction in the degree, respectively, and 2455 and 1700 for the betweenness method, which is equivalent to 24% and 12% reduction in the degree, respectively. The difference between the two methods is 21% for the blog network and 27% for the Wikipedia network, respectively. The worst contamination degree at 0.63% link block for the blog network, *i.e.*, $K = 500$ is 1045 for the proposed method and 1193 for the betweenness method, which is

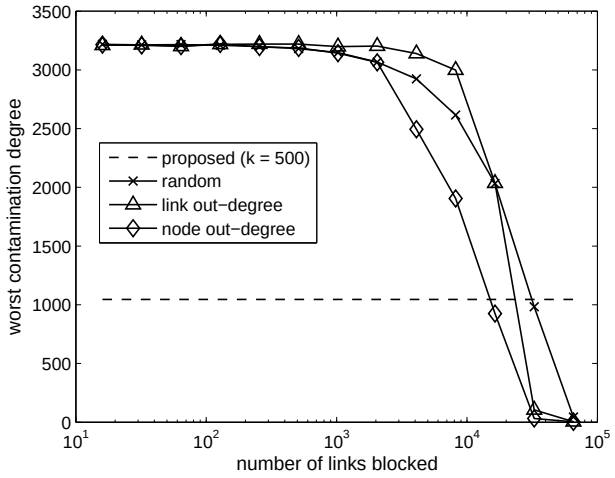


Fig. 11. Performance comparison of the proposed method for $K = 500$ with the node out-degree, link out-degree and random methods in the blog network for the worst contamination minimization problem.

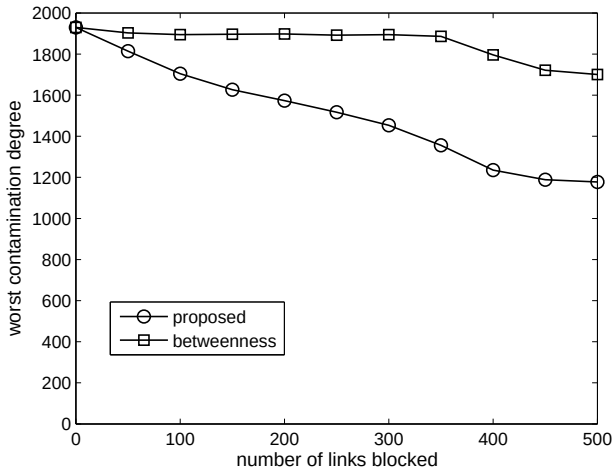


Fig. 12. Performance comparison between the proposed and betweenness methods in the Wikipedia network for the worst contamination minimization problem.

equivalent to 78% and 63% reduction in the degree, respectively, and the difference between the two methods is 15%.

Again differently from the above, the proposed method as well as the betweenness method outperform by far the other three methods (the node out-degree, the link out-degree and the random) for both the blog and the Wikipedia networks. Blocking 500 links by the proposed method is equivalent to blocking more than 10,000 links

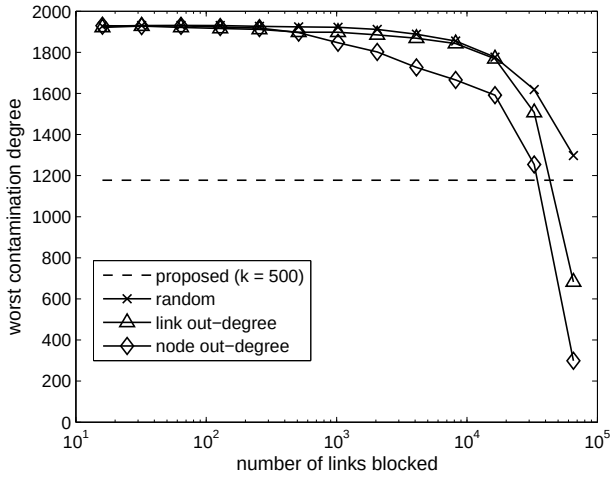


Fig. 13. Performance comparison of the proposed method for $K = 500$ with the node out-degree, link out-degree and random methods in the Wikipedia network for the worst contamination minimization problem.

for the blog network and 30,000 links by the other three methods, meaning that the proposed method is 20 to 60 times more effective.

5.4.3 *Discussion.* These results imply that the proposed method works effectively as expected, and outperforms the conventional link-removal heuristics. There is no big difference in the comparative performance results between the two networks. For both of them, the betweenness method performs reasonably well but the other three methods (the node out-degree, the link out-degree and the random) perform very poorly. There is no out-degree myth observed.

Of course how each of the conventional link-heuristics performs depends on the characteristics of the network structure. In general a network consists of multiple communities, and the members of each community are tightly connected and the members of different communities are less tightly connected. Thus, it is reasonable to assume that blocking the links between the different communities is effective in suppressing the contaminant to diffuse from one community to others. This is particularly true when there is a small number of nodes that play a key role of connecting different communities. Blocking these small number of paths is quite effective. The fact that the betweenness method performed reasonably well implies that the networks we analyzed may have this type of community structure. On the other hand, if the network is hierarchically structured, blocking the nodes, equivalently blocking the links attached to them, in the upper hierarchy should be quite effective. The fact that the node out-degree method does not do well suggests that there may not be such a structure in the networks we analyzed. Among the poorly performing three methods, the link out-degree method performs most poorly. It performs worse than the random methods for the blog network. This would indicate that it is mainly blocking the links within the communities.

With all these different factors affecting the performance of each method taken, the proposed method exhibits its strength of explicitly minimizing the contamination by considering the dynamics of information diffusion process, thereby making its performance less sensitive to the structure of the network.

Considering the fact that all the methods can eventually block the contamination when all of the links are blocked, it is important to have a method which is effective when the number of links to be blocked is limited to be small, and the proposed method has this property. It is noticeable that blocking only 0.2% of the links by the proposed method can reduce the contamination by nearly 50%.

We have devised two measures: the average contamination degree and the worst contamination degree. It is expected that the performance difference between the proposed method and the betweenness method is larger for the latter than the former, and the results is consistent. Our formulation does not assume the origins of contamination to be known and fixed. If they are known in advance, the problem is much easier computationally.

6. CONCLUSION

Just as good things, *e.g.*, innovation, important topics, etc. spread through a network and bring positive affects to people, undesirable things, e.g., computer virus, malicious rumors, etc. also spread and affect people badly. We addressed the problem of minimizing the spread of undesirable things by blocking links in a social network, which is converse to the influence maximization problem for the same network. In particular, we have considered two contamination minimization problems, one minimizing the average contamination degree and the other minimizing the worst (maximum) contamination degree. We chose to block "links" rather than "nodes" because deleting nodes necessitates deleting links, but not vice versa.

We have proposed novel methods for efficiently finding good approximate solutions to these problems on the basis of a naturally greedy algorithm and the bond percolation method. Using large-scale blog and Wikipedia networks, we have experimentally demonstrated that the proposed method works effectively, and also outperforms the conventional link-removal heuristics. The betweenness method performed reasonably well but the out-degree methods performed very poorly almost as badly as the random method. No out-degree myth was observed for the networks we analyzed. The performance of the link-removal heuristics is strongly affected by the network structure, but the proposed method shows that it is important to explicitly minimize the contamination by considering the dynamics of information diffusion process, which would make the performance less sensitive to the structure of the network.

ACKNOWLEDGMENTS

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

REFERENCES

Albert, R., Jeong, H., and Barabási, A.-L. 2000. Error and attack tolerance of complex networks. *Nature 406*, 378–382.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. 2000. Graph structure in the web. In *Proceedings of the 9th International World Wide Web Conference*. 309–320.

Callaway, D. S., Newman, M. E. J., Strogatz, S. H., and Watts, D. J. 2000. Network robustness and fragility: Percolation on random graphs. *Physical Reveiw Letters 85*, 5468–5471.

Domingos, P. and Richardson, M. 2001. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 57–66.

Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. 2004. Information diffusion through blogspace. In *Proceedings of the 13th International World Wide Web Conference*. 107–117.

Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 137–146.

Kimura, M., Saito, K., and Motoda, H. 2008. Minimizing the spread of contamination by blocking links in a network. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*. 1175–1180.

Kimura, M., Saito, K., and Nakano, R. 2007. Extracting influential nodes for information diffusion on a social network. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*. 1371–1376.

Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 420–429.

Newman, M. E. J. 2003. The structure and function of complex networks. *SIAM Review 45*, 167–256.

Newman, M. E. J., Forrest, S., and Balthrop, J. 2002. Email networks and the spread of computer viruses. *Physical Review E 66*, 035101.

Newman, M. E. J. and Girvan, M. 2004. Finding and evaluating community structure in networks. *Physical Review E 69*, 026113.

Newman, M. E. J. and Park, J. 2003. Why social networks are different from other types of networks. *Physical Review E 68*, 036122.

Richardson, M. and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 61–70.

Finding Influential Nodes in a Social Network from Information Diffusion Data

Masahiro Kimura[1], Kazumi Saito[2], Ryohei Nakano[3], and Hiroshi Motoda[4]

[1]kimura@rins.ryukoku.ac.jp, Ryukoku University, Shiga, Japan
[2]k-saito@u-shizuoka-ken.ac.jp, University of Shizuoka, Shizuoka, Japan
[3]nakano@cs.chubu.ac.jp, Chubu University, Aichi, Japan
[4]motoda@ar.sanken.osaka-u.ac.jp, Osaka Univesity, Osaka, Japan

Abstract We address the problem of ranking influential nodes in complex social networks by estimating diffusion probabilities from observed information diffusion data using the popular independent cascade (IC) model. For this purpose we formulate the likelihood for information diffusion data which is a set of time sequence data of active nodes and propose an iterative method to search for the probabilities that maximizes this likelihood. We apply this to two real world social networks in the simplest setting where the probability is uniform for all the links, and show that the accuracy of the probability is outstandingly good, and further show that the proposed method can predict the high ranked influential nodes much more accurately than the well studied conventional four heuristic methods.

1 Introduction

Innovation, hot topics and even malicious rumors can propagate through social networks among people in the form of so-called "word-of-mouth" communications. The rise of the Internet and the World Wide Web accelerates the creation of various large-scale social networks. Therefore, considerable attention has recently been devoted to social networks as an important medium for the spread of information.

Previous work addressed the problem of tracking the propagation patterns of topics or influence through blogspace [1, 5, 10], and studied strategies for removing nodes to prevent the spread of some undesriable information through a network, for example, the spread of a computer virus through an email network [2, 11]. A widely-used fundamental probabilistic model of information diffusion through a network is the *independent cascade (IC) model* [6, 5]. Using this model, the problem of finding a limited number of nodes that are effective for the spread of information [6, 8] have been extensively investigated. This combinatorial optimization problem is called the influence maximization problem. This problem was also investigated in a different setting (a descriptive probabilistic model of interaction) [4, 13]. Further, yet another problem of minimizing the spread of undesirable information by blocking a limited number of links in a network [9] has recently been addressed. In this paper, we also explore information diffusion phenomena for the IC model in a given network.

Overall, finding influential nodes in a social network is one of the most central problems in the field of social network analysis. There exist several methods for ranking nodes on the basis of the network structure [15]. We also address this problem, but from a different angle. We propose a method for extracting influential nodes by ranking nodes in terms of *influence degrees* for the IC model on the basis of the observed data of information diffusion in the network. The IC model is equipped with parameters. More specifically, the *diffusion probability* must be specified for each link in the network in advance. We estimate the probabilities so that the likelihood of obtaining the observed set of information diffusion data is maximized by an iterative algorithm (EM algorithm). Using two real world networks: the blog and Wikipedia networks, we first evaluate the accuracy of the diffusion probabilities and then use the estimated model to find the influential nodes and compare the results with the ground truth as well as the results that are obtained by using four strategies, each with a different heuristic, showing that the proposed method far outperforms the conventional methods.

The rest of the paper is organized as follows. The proposed method is formulated as a machine learning problem in section 2, and the experimental results together with the experimental settings are given in section 3, followed by some discussion of how the probabilities affect the influential nodes in section 4. We conclude this paper by summarizing our findings in section 5.

2 Proposed Method

2.1 Problem Formulation and Extraction Method

For a given directed network (or equivalently graph) $G = (V,E)$, let V be a set of nodes (or vertices) and E a set of links (or edges), where we denote each link by $e = (v,w) \in E$ and $v \neq w$, meaning there exists a directed link from a node v to a node w. For each node v in the network G, we denote $F(v)$ as a set of child nodes of v as follows: $F(v) = \{w; (v,w) \in E\}$. Similarly, we denote $B(v)$ as a set of parent nodes of v as follows: $B(v) = \{u; (u,v) \in E\}$.

In the IC model, for each directed link $e = (v,w)$, we specify a real value $p_{v,w}$ with $0 < p_{v,w} < 1$ in advance. Here $p_{v,w}$ is referred to as the *diffusion probability* of link (v,w). The diffusion process proceeds from a given initial active set $D(0)$ in the following way. When a node v first becomes active at time-step t, it is given a single chance to activate each currently inactive child node w, and succeeds with probability $p_{v,w}$. If v succeeds, then w will become active at time-step $t+1$. If multiple parent nodes of w first become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not v succeeds, it cannot make any further attempts to activate w in subsequent rounds. The process terminates if no more activations are possible.

For a given set of diffusion probabilities, $\Theta = \{p_{v,w}; (v,w) \in E\}$, and an initial active node v, we define the *influence degree*, denoted by $\sigma(v;\Theta)$, as the expected number of active nodes. Our problem of finding influential nodes is formulated as a node ranking problem based on the influence degree $\sigma(v;\Theta_0)$, where Θ_0 means a set of the true diffusion probabilities. In practice settings, however, the true diffusion probability set Θ_0 is not available. Thus, we consider to utilize their probabilities $\widehat{\Theta}$ estimated from past information diffusion histories observed as sets of active nodes. Then we need to evaluate the ranking similarity between two sorted node lists according to $\sigma(v;\Theta_0)$ and $\sigma(v;\widehat{\Theta})$.

2.2 Probability Estimation Method

Let $D = D(0) \cup D(1) \cup \cdots \cup D(T)$ be an information diffusion result, where $D(t)$ is the set of nodes that have become active at time t. When $v \in D(t)$ and $w \in D(t+1) \cap F(v)$ hold for some link $e = (v,w)$, it is possible that the node v succeeded in activating the node w via the link e. However, since we should consider possibilities that some other nodes $v' \in D(t) \cap B(w)$ also succeeded in activating the node w, we need to calculate the probability that the node w becomes active at time $t+1$ as follows: $P(w;t+1) = 1 - \prod_{v \in B(w) \cap D(t)} (1 - p_{v,w})$. Here note that if $w \in D(t+1)$, it is guaranteed that $D(t) \cap B(w) \neq \emptyset$.

We set $C(t) = D(0) \cup \cdots \cup D(t)$. Note that $C(t)$ is the set of active nodes at time t. When $v \in D(t)$ and $w \in F(v) \setminus C(t+1)$ hold, we know that the node v definitely failed to activate the node w via the link e. Clearly, when $v \in D(t)$ and $w \in F(v) \cap C(t)$ hold, as well as $v \notin D$, no information is available about the trial with respect to the link $e = (v,w)$. Therefore, we can define the likelihood function with respect to $\Theta = \{p_{v,w}\}$ as follows:

$$\mathscr{L}(\Theta;D) = \prod_{t=0}^{T-1} \prod_{w \in D(t+1)} \left(1 - \prod_{v \in B(w) \cap D(t)} (1 - p_{v,w})\right) \prod_{t=0}^{T} \prod_{v \in D(t)} \prod_{w \in F(v) \setminus C(t+1)} (1 - p_{v,w}).$$

Let $\{D_m; 1 \leq m \leq M\}$ be an observed data set of M independent information diffusion results. Then we can define the following objective function with respect to Θ:

$$\mathscr{J}(\Theta) = \sum_{m=1}^{M} \log \mathscr{L}(\Theta;D_m). \tag{1}$$

Thus, our problem is to obtain the set of information diffusion probabilities Θ, which maximizes Equation (1). For this estimation problem, we have already proposed an estimation method based on the Expectation-Maximization algorithm in order to stably obtain its solutions [14].

In order to evaluate fundamental abilities of our method, in this paper, we consider the simplest case that all links have the same diffusion probability p. Note that this problem setting has been widely adopted in many previous experiments

[6, 8, 9], and the formulation is valid for more general cases in which there is no such restriction.

3 Experiments

3.1 Experimental Settings

We employed two sets of large real networks used in [9], the blog and Wikipedia networks, which exhibit many of the key features of social networks. These are bidirectional networks. The blog network had $12,047$ nodes and $79,920$ directed links, and the Wikipedia network had $9,481$ nodes and $245,044$ directed links. As stated before, in our preliminary experiments, we assumed the simplest case where the diffusion probability is uniform throughout the network, and set the value p as follows: $p = 0.1$ for the blog network and $p = 0.01$ for the Wikipedia network. We evaluated the influence degrees $\{\sigma(v);\ v \in V\}$ using the method of [8] with the parameter value $10,000$, where the parameter represents the number of bond percolation processes (we do not describe the method here due to the page limit). The average value and the standard deviation of the influence degrees was 87.5 and 131 for the blog network, and 8.14 and 18.4 for the Wikipedia network.

In the learning stage, a training sample was an information diffusion path $D = D(0) \cup D(1) \cup \cdots \cup D(T)$ which is a sequence of the active nodes starting from a randomly selected initial active node. We used M training samples for learning the propagation probability, where M is a parameter.

3.2 Comparison Methods

We compared the proposed method with four heuristics from social network analysis with respect to the predictive capability of high ranked influential nodes.

First, "degree centrality", "closeness centrality", and "betweenness centrality" are commonly used as influence measure in sociology [15], where the degree of node v is defined as the number of links attached to v, the closeness of node v is defined as the reciprocal of the average distance between v and other nodes in the network, and the betweenness of node v is defined as the total number of shortest paths between pairs of nodes that pass through v.

We also consider measuring the influence of each node by its "authoritativeness" obtained by the "PageRank" method [3], since this is a well known method for identifying authoritative or influential pages in a hyperlink network of web pages. This method has a parameter ε; when we view it as a model of a random web surfer, ε corresponds to the probability with which a surfer jumps to a page picked uniformly at random [12]. In our experiments, we used a typical setting of $\varepsilon = 0.15$.

3.3 Experimental Results

First, we examined the learning performance of propagation probability by the proposed method. Let p_0 be the true value of propagation probability, and let $\hat{p}$ be the value of propagation probability estimated by the proposed method. We evaluated the learning performance in terms of the error rate $\mathscr{E} = |p_0 - \hat{p}|/p_0$.

Table 1 Learing performance of propagation probability.

Results for the blog network

M	$\mathscr{E}$
20	0.036 (0.024)
40	0.018 (0.014)
60	0.016 (0.007)
80	0.009 (0.006)
100	0.006 (0.004)

Results for the Wikipedia network

M	$\mathscr{E}$
20	0.138 (0.081)
40	0.109 (0.066)
60	0.080 (0.041)
80	0.047 (0.018)
100	0.021 (0.013)

Table 1 shows the average value of $\mathscr{E}$ and the standard deviation in parenthesis for the number of training samples, M, where we performed the same experiment five times independently. Our algorithm can converge to the true value efficiently when there is a reasonable amount of training data. The results are better for a larger value of diffusion probability. The results demonstrate the effectiveness of the proposed method.

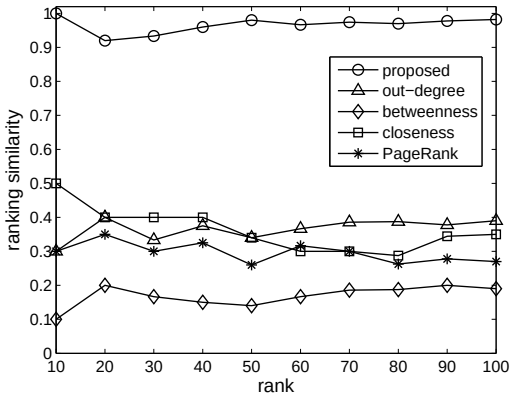


Fig. 1 Performance comparison in extracting influential nodes for the blog network.

Next, in terms of ranking for extracting influential nodes from the network $G = (V,E)$, we compared the proposed method with the out-degree, the betweenness, the closeness, and the PageRank methods. For any positive integer r $(\leq |V|)$,

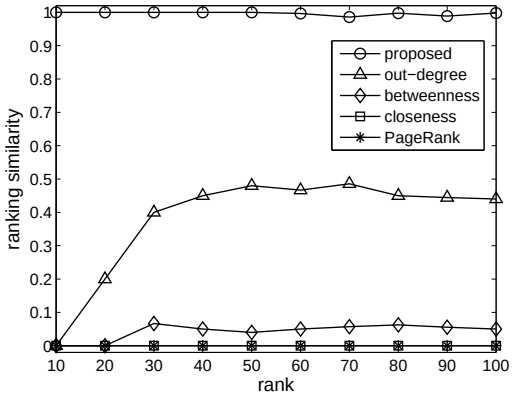


Fig. 2 Performance comparison in extracting influential nodes for the Wikipedia network.

let $L_0(r)$ be the true set of top r nodes, and let $L(r)$ be the set of top r nodes for a given ranking method. We evaluated the performance of the ranking method by the *ranking similarity* $F(r)$ at rank r, where $F(r)$ is defined by $F(r) = |L_0(r) \cap L(r)|/r$. We focused on ranking similarities at high ranks since we are interested in extracting influential nodes. Figures 1 and 2 show the results for the blog and the Wikipedia networks, respectively. Here, circles, triangles, diamonds, squares, and asterisks indicate ranking similarity $F(r)$ as a function of rank r for the proposed, the out-degree, the betweenness, the closeness, and the PageRank methods, respectively. For the proposed method, we plotted the average value of $F(r)$ at r for five experimental results in the case of $M = 100$. The proposed method gives far better results than the other heuristic based methods for the both networks demonstrating the effectiveness of the proposed method.

4 Discussion

We consider that our proposed ranking method presents a novel concept of centrality based on the information diffusion model, i.e., *the IC model*. Actually, Figures 1 and 2 show that nodes identified as higher ranked by our method are substantially different from those by each of the conventional methods. This means that our method enables a new type of social network analysis if past information diffusion data are available. Of course, it is beyond controversy that each conventional method has its own merit and usage, and our method is an addition to them which has a different merit in terms of information diffusion.

Here, we do some simple analysis of explaining why it is important to know the diffusion probability in finding the influential nodes. If the probability does not affect the ranking, we don't care about its absolute value. However, a simple anal-

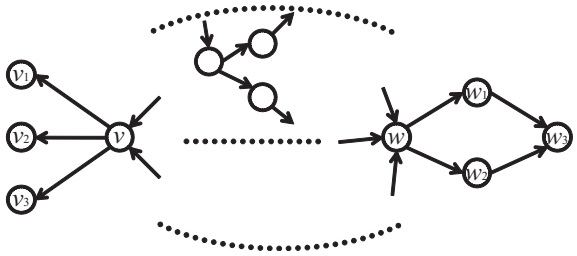


Fig. 3 An example of network.

ysis reveals that it does affect the node ranking. Note that $\sigma(v;p)$ is a monotonically increasing non negative function of p if v's out degree is non zero. Assume that there are two such nodes v and w that have the following graph structures: $(v,v_1),(v,v_2),(v,v_3) \in E$ and $(w,w_1),(w,w_2),(w_1,w_3),(w_2,w_3) \in E$ (see Fig. 3). The maximum influential degree is 3 for the both nodes v and w. The expected values are easily calculated [7] as $\sigma(v;p) = 3p$, and $\sigma(w;p) = 2p + (1-(1-p^2)^2) = 2p + 2p^2 - p^4$. Thus, $\sigma(v;p) - \sigma(w;p) = p(1-p)(1-p-p^2)$. From this, if $p < (-1+\sqrt{5})/2, \sigma(v;p) > \sigma(w;p)$. Otherwise, $\sigma(v;p) \leq \sigma(w;p)$. Intuitively, as p gets larger, the influential probability of the nodes reachable in two steps from the starting node becomes larger than that of the nodes reachable in one step, and thus, w that has child nodes in two steps downward has a larger influential degree. Since in general there are many subnetworks like these within a network, it is important to estimate the diffusion probabilities as accurately as possible. We believe that the methods proposed in this paper would contribute to various types of social network analyses.

We note that the analysis we showed in this paper is the simplest case where p takes a single value for all the links in E. However, the method is very general. In a more realistic setting we can divide E into subsets $E_1, E_2, ..., E_N$ and assign a different value p_n for all the links in each E_n. For example, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. If there is some background knowledge about the node grouping, our method can make the best use of it, one of the characteristics of the artificial intelligence approach. Obtaining such background knowledge is also an important research topic in the knowledge discovery from social networks.

5 Conclusion

We addressed the problem of ranking influential nodes in complex social networks, given the network topology and the observation data of information diffusion. We

formulated how to estimate the diffusion probability of each link from the past information diffusion histories observed as sets of active nodes using the popular information diffusion model, *IC model* as a likelihood maximization problem and derived an efficient iterative EM method to solve it. The results we obtained by applying to two real world networks in the simplest setting where the probability is uniform throughout each network show that 1) the method can estimate the probability accurately when there is enough number of observation sequence data that can be used for training and 2) the ranking of influential nodes predicted by the method far outperforms the other well known heuristic based methods (degree centrality, closeness centrality, betweenness centrality, and authoritativeness).

Acknowledgements This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Adar E, Adamic L (2005) Tracking information epidemics in blogspace. In: WI'05 207–214
2. Albert R, Jeong H, Barabási A L (2000) Error and attack tolerance of complex networks. Nature 406:378–382
3. Brin S, Page L (1998) The anatomy of a large-scale hypertextual web search engine. In: WWW'98 107–117
4. Domingos P, Richardson M (2001) Mining the network value of customers. In: KDD'01 57–66
5. Gruhl D, Guha R, Liben-Nowell D, Tomkins A (2004) Information diffusion through blogspace. In: WWW'04 107–117
6. Kempe D, Kleinberg J, Tardos E (2003) Maximizing the spread of influence through a social network. In: KDD'03 137–146
7. Kimura M, Saito K (2006) Tractable models for information diffusion in social networks. In: PKDD'06 259–271
8. Kimura M, Saito K, Nakano R (2007) Extracting influential nodes for information diffusion on a social network. In: AAAI'07 1371–1376
9. Kimura M, Saito K, Motoda H (2008) Minimizing the spread of contamination by blocking links in a network. In: AAAI'08 1175–1180
10. Leskovec J, Adamic L, Huberman B A (2006) The dynamics of viral marketing. In: EC'06 228–237
11. Newman M E J, Forrest S, Balthrop J (2002) Email networks and the spread of computer viruses. Phys Rev E 66:035101
12. Ng A Y, Zheng A X, Jordan M I (2001) Link analysis, eigenvectors and stability. In: IJCAI'01 903–901
13. Richardson M, Domingos P (2002) Mining knowledge-sharing sites for viral marketing. In: KDD'02 61–70
14. Saito K, Nakano R, Kimura M (2008) Prediction of information diffusion probabilities for independent cascade model. In: KES'08 67–75
15. Wasserman S, Faust K (1994) Social network analysis. Cambridge University Press, Cambridge, UK

Efficient Estimation of Influence Functions for SIS Model on Social Networks*

Masahiro Kimura
Department of Electronics and Informatics
Ryukoku University
kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics
University of Shizuoka
k-saito@u-shizuoka-ken.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research
Osaka University
motoda@ar.sanken.osaka-u.ac.jp

Abstract

We address the problem of efficiently estimating the influence function of initially activated nodes in a social network under the *susceptible/infected/susceptible (SIS) model*, a diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property. We solve this problem by constructing a layered graph from the original social network with each layer added on top as the time proceeds, and applying the bond percolation with a pruning strategy. We show that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis and confirm this by applying the proposed method to two real world networks.

1 Introduction

Social networks mediate the spread of various information including topics, ideas and even (computer) viruses. The proliferation of emails, blogs and social networking services (SNS) in the World Wide Web accelerates the creation of large social networks. Therefore, substantial attention has recently been directed to investigating information diffusion phenomena in social networks [Adar and Adamic, 2005; Leskovec *et al.*, 2007b; Agarwal and Liu, 2008].

Overall, finding influential nodes is one of the most central problems in social network analysis. Thus, developing methods to do this on the basis of information diffusion is an important research issue. Widely-used fundamental probabilistic models of information diffusion are the *independent cascade (IC) model* and the *linear threshold (LT) model* [Kempe *et al.*, 2003; Gruhl *et al.*, 2004]. Researchers investigated the problem of finding a limited number of influential nodes that are effective for the spread of information under the above models [Kempe *et al.*, 2003; Kimura *et al.*, 2007]. This combinatorial optimization problem is called the *influence maximization problem*. Kempe *et al.* [2003] experimentally showed on large collaboration networks that the greedy algorithm can give a good approximate solution to this problem, and mathematically proved a performance guarantee of the greedy solution (i.e., the solution obtained by the greedy algorithm). Recently, methods based on bond percolation [Kimura *et al.*, 2007] and submodularity [Leskovec *et al.*, 2007a] were proposed for efficiently estimating the greedy solution. The influence maximization problem has applications in sociology and "viral marketing" [Agarwal and Liu, 2008], and was also investigated in a different setting (a descriptive probabilistic model of interaction) [Domingos and Richardson, 2001; Richardson and Domingos, 2002]. The problem has recently been extended to influence control problems such as a contamination minimization problem [Kimura *et al.*, 2009].

The IC model can be identified with the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease [Newman, 2003; Gruhl *et al.*, 2004]. In the SIR model, only infected individuals can infect susceptible individuals, while recovered individuals can neither infect nor be infected. This implies that an individual is never infected with the disease multiple times. This property holds true for the LT model as well. However, there exist phenomena for which the property does not hold. For example, consider the following propagation phenomenon of a topic in the blogosphere: A blogger who has not yet posted a message about the topic is interested in the topic by reading the blog of a friend, and posts a message about it (i.e., becoming infected). Next, the same blogger reads a new message about the topic posted by some other friend, and may post a message (i.e., becoming infected) again. Most simply, this phenomenon can be modeled by an *susceptible/infected/susceptible (SIS) model* from the epidemiology. Like this example, there are many examples of information diffusion phenomena for which the SIS model is more appropriate, including the growth of hyper-link posts among bloggers [Leskovec *et al.*, 2007b], the spread of computer viruses without permanent virus-checking programs, and epidemic disease such as tuberculosis and gonorrhea [Newman, 2003]. In this paper, we focus on an information diffusion process in a social network over a given time span on the basis of an SIS model.

Here, the SIS model is a stochastic process model, and the *influence* of a node v at time-step t, $\sigma(v,t)$, is defined as the expected number of infected nodes at time-step t when v is

*This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and Grant-in-Aid for Scientific Research (C) (No. 20500147).

initially infected at time-step $t = 0$. We refer to σ as the *influence function* for the SIS model. Developing an effective method for estimating σ is vital for various applications. Clearly, in order to extract influential nodes, we must estimate the value of $\sigma(v, t)$ for every node v and time-step t. Moreover, note that the method developed can be easily extended and applied to approximately solving the influence maximization problem for the SIS model by the greedy alogrithm. We can naively estimate σ by simulating the SIS model. However, this naive method is overly inefficient and not practical at all as shown in the experiments. In this paper, we propose a method for estimating influence function σ efficiently. By theoretically comparing computational complexity with the naive method, we show that the proposed method is expected to achieve a large reduction in computational cost. Further, using two large real networks, we experimentally demonstrate that the proposed method is much more efficient than the naive method with the same accuracy.

2 Information Diffusion Model

Let $G = (V, E)$ be a directed network, where V and E ($\subset V \times V$) stand for the sets of all the nodes and (directed) links, respectively. For any $v \in V$, let $\Gamma(v; G)$ denote the set of the child nodes (directed neighbors) of v, that is,

$$\Gamma(v; G) = \{w \in V;\ (v, w) \in E\}.$$

2.1 SIS Model

An SIS model for the spread of a disease is based on the cycle of disease in a host. A person is first *susceptible* to the disease, and becomes *infected* with some probability when the person encounters an infected person. The infected person becomes susceptible to the disease soon without moving to the immune state. We consider a discrete-time SIS model for information diffusion on a network. In this context, infected nodes mean that they have just adopted the information, and we call these infected nodes *active* nodes.

We define the SIS model for information diffusion on G. In the model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that the state of a node is either active or inactive. For every link $(u, v) \in E$, we specify a real value $p_{u,v}$ with $0 < p_{u,v} < 1$ in advance. Here, $p_{u,v}$ is referred to as the *propagation probability* through link (u, v). Given an initial set of active nodes X and a time span T, the diffusion process proceeds in the following way. Suppose that node u becomes active at time-step t ($< T$). Then, node u attempts to activate every $v \in \Gamma(u; G)$, and succeeds with probability $p_{u,v}$. If node u succeeds, then node v will become active at time-step $t + 1$. If multiple active nodes attempt to activate node v in time-step t, then their activation attempts are sequenced in an arbitrary order. On the other hand, node u will become inactive at time-step $t + 1$ unless it is activated from an active node in time-step t. The process terminates if the current time-step reaches the time limit T.

2.2 Influence Function

For the SIS model on G, we consider a diffusion sample from an initial active node $v \in V$ over time span T. Let $S(v, t)$ denote the set of active nodes at time-step t. Note that $S(v, t)$ is a random subset of V and $S(v, 0) = \{v\}$. Let $\sigma(v, t)$ denote the expected number of $|S(v, t)|$, where $|X|$ stands for the number of elements in a set X. We call $\sigma(v, t)$ the *influence* of node v at time-step t. Note that σ is a function defined on $V \times \{0, 1, \cdots, T\}$. We call the function σ the *influence function* for the SIS model over time span T on network G.

It is important to estimate the influence function σ efficiently. We can simply estimate σ by the simulations based on the SIS model in the following way. First, a sufficiently large positive integer M is specified. For each $v \in V$, the diffusion process of the SIS model is simulated from the initial active node v, and the number of active nodes at time-step t, $|S(v, t)|$, is calculated for every $t \in \{0, 1, \cdots, T\}$. Then, $\sigma(v, t)$ is estimated as the empirical mean of $|S(v, t)|$'s that are obtained from M such simulations. We refer to this estimation method as the *naive method*. As shown in the experiments, the naive method is extremely inefficient, and cannot be practical.

3 Proposed Method

We propose a method for efficiently estimating the influence function σ over time span T for the SIS model on network G.

3.1 Layered Graph

We build a layered graph $G^T = (V^T, E^T)$ from G in the following way. First, for each node $v \in V$ and each time-step $t \in \{0, 1, \cdots, T\}$, we generate a copy v_t of v at time-step t. Let V_t denote the set of copies of all $v \in V$ at time-step t. We define V^T by $V^T = V_0 \cup V_1 \cup \cdots \cup V_T$. In particular, we identify V with V_0. Next, for each link $(u, v) \in E$, we generate T links (u_{t-1}, v_t), ($t \in \{1, \cdots, T\}$), in the set of nodes V^T. We set $E_t = \{(u_{t-1}, v_t); (u, v) \in E\}$, and define E^T by $E^T = E_1 \cup \cdots \cup E_T$. Moreover, for any link (u_{t-1}, v_t) of the layered graph G^T, we define the occupation probability q_{u_{t-1},v_t} by $q_{u_{t-1},v_t} = p_{u,v}$.

Then, we can easily prove that the SIS model with propagation probabilities $\{p_e; e \in E\}$ on G over time span T is equivalent to the *bond percolation process (BP) with occupation probabilities* $\{q_e; e \in E^T\}$ on G^T.[1] Here, the BP process with occupation probabilities $\{q_e; e \in E^T\}$ on G^T is the random process in which each link $e \in E^T$ is independently declared "occupied" with probability q_e. We perform the BP process on G^T, and generate a graph constructed by occupied links, $\tilde{G}^T = (V^T, \tilde{E}^T)$. Then, in terms of information diffusion by the SIS model on G, an occupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information propagates at time-step t, and an unoccupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information does not propagate at time-step t. For any $v \in V$, let $F(v; \tilde{G}^T)$ be the set of all nodes that can be reached from v (= v_0) through a path on the graph $\tilde{G}^T$. When we consider a diffusion sample from an initial active node $v \in V$ for the SIS model on G, $F(v; \tilde{G}^T) \cap V_t$ represents the set of active nodes at time-step t, $S(v, t)$.

[1] The SIS model over time span T on G can be exactly mapped onto the IC model on G^T [Kempe *et al.*, 2003]. Thus, the result follows from the equivalence of the BP process and the IC model [Newman, 2003; Kempe *et al.*, 2003; Kimura *et al.*, 2007].

3.2 Bond Percolation Method

Using the equivalent BP process, we present a method for efficiently estimating influence function σ. We refer to this method as the *BP method*. Unlike the naive method, the BP method simultaneously estimates $\sigma(v,t)$ for all $v \in V$. Moreover, the BP method does not fully perform the BP process, but performs it partially. Note first that all the paths from a node $v \in V$ on the graph $\tilde{G}^T$ represent a diffusion sample from the initial active node v for the SIS model on G. Let L' be the set of the links in G^T that is not in the diffusion sample. For calculating $|S(v,t)|$, it is unnecessary to determine whether the links in L' are occupied or not. Therefore, the BP method performs the BP process for only an appropriate set of links in G^T. The BP method estimates σ by the following algorithm:

BP method:

1. Set $\sigma(v,t) \leftarrow 0$ for each $v \in V$ and $t \in \{1,\cdots,T\}$.

2. Repeat the following procedure M times:

2-1. Initialize $S(v,0) = \{v\}$ for each $v \in V$, and set $A(0) \leftarrow V$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.

2-2. For $t = 1$ to T do the following steps:

2-2a. Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(v,t-1)$.

2-2b. Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

2-2c. For each $v \in A(t-1)$, compute $S(v,t) = \bigcup_{w \in S(v,t-1)} \Gamma(w; \tilde{G}_t)$, and set $\sigma(v,t) \leftarrow \sigma(v,t) + |S(v,t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(v,t) \neq \emptyset$.

3. For each $v \in V$ and $t \in \{1,\cdots,T\}$, set $\sigma(v,t) \leftarrow \sigma(v,t)/M$, and output $\sigma(v,t)$.

Note that $A(t)$ finally becomes the set of information source nodes that have at least an active node at time-step t, that is, $A(t) = \{v \in V;\, S(v,t) \neq \emptyset\}$. Note also that $B(t-1)$ is the set of nodes that are activated at time-step $t-1$ by some source nodes, that is, $B(t-1) = \bigcup_{v \in V} S(v,t-1)$.

Now we estimate the computational complexity of the BP method in terms of the number of the nodes, $\mathcal{N}_a$, that are identified in step 2-2a, the number of the coin-flips, $\mathcal{N}_b$, for the BP process in step 2-2b, and the number of the links, $\mathcal{N}_c$, that are followed in step 2-2c. Let $d(v)$ be the number of out-links from node v (i.e., out-degree of v) and $d'(v)$ the average number of occupied out-links from node v after the BP process. Here we can estimate $d'(v)$ by $\sum_{w \in \Gamma(v;G)} p_{v,w}$. Then, for each time-step $t \in \{1,\cdots,T\}$, we have

$$\mathcal{N}_a = \sum_{v \in A(t-1)} |S(v,t-1)|, \quad \mathcal{N}_b = \sum_{w \in B(t-1)} d(w), \quad (1)$$

and

$$\mathcal{N}_c = \sum_{v \in A(t-1)} \sum_{w \in S(v,t-1)} d'(w) \quad (2)$$

on average.

In order to compare the computational complexity of the BP method to that of the naive method, we consider mapping the naive method onto the BP framework, that is, separating the coin-flip process and the link-following process. We can easily verify that the following algorithm in the BP framework is equivalent to the naive method:

A method that is equivalent to the naive method:

1. Set $\sigma(v,t) \leftarrow 0$ for each $v \in V$ and $t \in \{1,\cdots,T\}$.

2. Repeat the following procedure M times:

2-1. Initialize $S(v,0) = \{v\}$ for each $v \in V$, and set $A(0) \leftarrow V$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.

2-2. For $t = 1$ to T do the following steps:

2-2b'. For each $v \in A(t-1)$, perform the BP process for the links from $S(v,t-1)$ in G^T, and generate the graph $\tilde{G}_t(v)$ constructed by the occupied links.

2-2c'. For each $v \in A(t-1)$, compute $S(v;t) = \bigcup_{w \in S(v,t-1)} \Gamma(w; \tilde{G}_t(v))$, and set $\sigma(v,t) \leftarrow \sigma(v,t) + |S(v,t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(v,t) \neq \emptyset$.

3. For each $v \in V$ and $t \in \{1,\cdots,T\}$, set $\sigma(v,t) \leftarrow \sigma(v,t)/M$, and output $\sigma(v,t)$.

Then, for each $t \in \{1,\cdots,T\}$, the number of coin-flips, $\mathcal{N}_{b'}$, in step 2-2b' is

$$\mathcal{N}_{b'} = \sum_{v \in A(t-1)} \sum_{w \in S(v,t-1)} d(w), \quad (3)$$

and the number of the links, $\mathcal{N}_{c'}$, followed in step 2-2c' is equal to $\mathcal{N}_c$ in the BP method on average. From equations (2) and (3), we can see that $\mathcal{N}_{b'}$ is much larger than $\mathcal{N}_{c'} = \mathcal{N}_c$, especially for the case where the diffusion probabilities are small. By equations (1) and (3), we can also see that $\mathcal{N}_{b'}$ is generally much larger than each of $\mathcal{N}_a$ and $\mathcal{N}_b$ in the BP method for a real social network. In fact, since such a network generally includes large clique-like subgraphs, there are many nodes $w \in V$ such that $d(w) \gg 1$, and we can expect that $\sum_{v \in A(t-1)} |S(v,t-1)| \gg |\bigcup_{v \in A(t-1)} S(v,t-1)|$ $(= |B(t-1)|)$. Therefore, the BP method is expected to achieve a large reduction in computational cost.

3.3 Pruning Method

In order to further improve the computational efficiency of the BP method, we introduce a pruning technique and propose a method referred to as the *BP with pruning method*. The key idea of the pruning technique is to utilize the following property: Once we have $S(u,t_0) = S(v,t_0)$ at some time-step t_0 on the course of the BP process for a pair of information source nodes, u and v, then we have $S(u,t) = S(v,t)$ for all $t > t_0$. The BP with pruning method estimates σ by the following algorithm:

BP with pruning method:

1. Set $\sigma(v,t) \leftarrow 0$ for each $v \in V$ and $t \in \{1,\cdots,T\}$.

2. Repeat the following procedure M times.

2-1". Initialize $S(v;0) = \{v\}$ for each $v \in V$, and set $A(0) \leftarrow V$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V$.

2-2. For $t = 1$ to T do the following steps:

2-2a. Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(v, t-1)$.

2-2b. Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

2-2c". For each $v \in A(t-1)$, compute $S(v,t) = \bigcup_{w \in S(v,t-1)} \Gamma(w; \tilde{G}_t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(v,t) \neq \emptyset$, and set $\sigma(u,t) \leftarrow \sigma(u,t) + |S(v,t)|$ for each $u \in C(v)$.

2-2d. Check whether $S(u,t) = S(v,t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(u,t) = S(v,t)$.

3. For each $v \in V$ and $t \in \{1, \cdots, T\}$, set $\sigma(v,t) \leftarrow \sigma(v,t)/M$, and output $\sigma(v,t)$.

Basically, by introducing step 2-2d and reducing the size of $A(t)$, the proposed method attempts to improve the computational efficiency in comparison to the original BP method.

For the proposed method, it is important to implement efficiently the equivalence check process in step 2-2d. In our implementation, we first classify each $v \in A(t)$ according to the value of $k = |S(v,t)|$, and then perform the equivalence check process only for those nodes with the same k value. How effectively the proposed method works will depend on several conditions such as network structure, time span, values of diffusion probabilities, and so on. We will do a simple analysis later and experimentally show that it is indeed effective.

4 Experimental Evaluation

4.1 Network Data and Settings

In our experiments, we employed two datasets of large real networks used in [Kimura *et al.*, 2009], which exhibit many of the key features of social networks.

The first one is a trackback network of Japanese blogs. The network data was collected by tracing the trackbacks from one blog in the site "goo (http://blog.goo.ne.jp/)" in May, 2005. We refer to the network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications by putting trackbacks on each other's blogs. The blog network had $12,047$ nodes and $79,920$ directed links.

The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. We refer to the network data as the Wikipedia network. The Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

For the SIS model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any link $(u,v) \in E$, that is, $p_{u,v} = p$. According to [Kempe *et al.*, 2003; Leskovec *et al.*, 2007b], we set the value of p relatively small. In particular, we set the value of p to a value smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. We decided to set $p = 0.1$ for the blog network and $p = 0.01$ for the Wikipedia network.

All our experimentation was undertaken on a single PC with an Intel Core 2 Duo E6850 3GHz processor, with 3GB of memory, running under Linux.

4.2 Estimation Accuracy Comparison

We first compared the accuracy of the estimated influence function σ of the proposed method (BP with pruning) with that of the naive method. Both methods require M to be specified in advance as a parameter. As shown in section 3.2, the number of coin flips is different in these two methods and it is much larger in the naive method. However, this does not mean that there is more randomness introduced in the naive method and thus the convergence of the naive method is faster. In fact for each single initially activated node v from which to propagate the information, the number of independent coin-flips is effectively the same for the both methods. Thus by using the same value of M, both would estimate $\sigma(v,t)$ with the same accuracy in principle.

Table 1: Results for the naive method on the blog network.

Rank	Node ID	Influence	Node ID	Influence
1	2210	984.38	2210	985.74
2	2248	979.59	2248	980.72
3	3906	956.82	3906	956.57
4	3907	953.14	3907	953.89
5	146	931.03	146	931.62
6	155	929.68	155	930.21
7	3233	913.50	3233	911.89
8	3228	912.27	3228	910.52
9	140	910.04	140	910.37
10	2247	909.59	2247	910.00

Table 2: Results for the proposed method on the blog network.

Rank	Node ID	Influence	Node ID	Influence
1	2210	984.74	2210	984.87
2	2248	980.41	2248	979.46
3	3906	956.97	3906	955.84
4	3907	953.04	3907	952.71
5	146	929.96	146	929.30
6	155	928.77	155	928.49
7	3233	912.61	3233	911.01
8	3228	912.18	3228	910.49
9	140	909.22	140	910.31
10	2247	909.12	2247	909.59

We have experimentally confirmed that use of $M = 100,000$ gives in effect the same value of $\sigma(v,t)$, for $t = 1, \cdots, 20$. The following accuracy comparison is based on $M = 100,000$. Tables 1 and 2 show the ranking of the influential initially activated nodes v evaluated at time-step $T = 20$ for the blog network. The value of influence function $\sigma(v, 20)$ is sorted in the decreasing order and the top 10 nodes are listed. We repeated the experiment several times and listed two of them. Note that the naive method takes an order of week to return the result and we could not set T a

Table 3: Results for the naive method on the Wikipedia network.

Rank	Node ID	Influence	Node ID	Influence
1	4019	134.73	4019	133.83
2	3729	133.24	3729	132.42
3	7919	132.66	7919	131.98
4	4380	132.23	1720	131.68
5	1720	132.20	4380	131.34
6	4465	132.10	4465	131.07
7	1712	131.65	1712	130.69
8	3670	130.32	1073	129.48
9	1073	129.66	3670	129.46
10	1191	128.61	1191	128.38

Table 4: Results for the proposed method on the Wikipedia network.

Rank	Node ID	Influence	Node ID	Influence
1	4019	134.25	4019	133.67
2	3729	132.91	7919	132.17
3	7919	132.50	3729	132.02
4	4380	132.03	4380	131.84
5	4465	131.95	1720	131.63
6	1720	131.59	4465	131.12
7	1712	131.33	1712	130.90
8	3670	130.27	3670	129.78
9	1073	129.22	1073	129.12
10	1191	128.71	1191	128.40

larger value. We note that the ranking is exactly the same for the both methods. Tables 3 and 4 are the result for the Wikipedia network. The nodes in the 4th and the 5th ranks for the naive method, and the 5th and the 6th ranks for the proposed method are interchanged respectively, but the rests are the same. From these results we confirm that the proposed method gives the same results as the naive method with the same value of M when M is large enough.

4.3 Processing Time Comparison

Next, we compared the processing time of the proposed method (BP with pruning) with the BP method without pruning and the naive method. Here, we used $M = 1,000$ in order to keep the computational time for the naive method at a reasonable level so that it runs for a larger T. Figures 1 and 2 show the total processing time to estimate $\{\sigma(v,t); v \in V, t = 0, 1, \cdots, T\}$ as a function of time span T for the blog and the Wikipedia networks, respectively. In these figures, the circles, squares and triangles indicate the results for the proposed method (BP with pruning), the BP method without pruning, and the naive method, respectively. Note that in case of the blog network, the processing time for time span $T = 100$ is about 7 minutues, 2 hours and 37 hours for the proposed method, the BP method without pruning and the naive method, respectively. Namely, the proposed method is about 20 and 310 times faster than the BP method without pruning and the naive method, respectively, for $T = 100$ in case of the blog network. Note also that in case of the Wikipedia network, the processing time for time

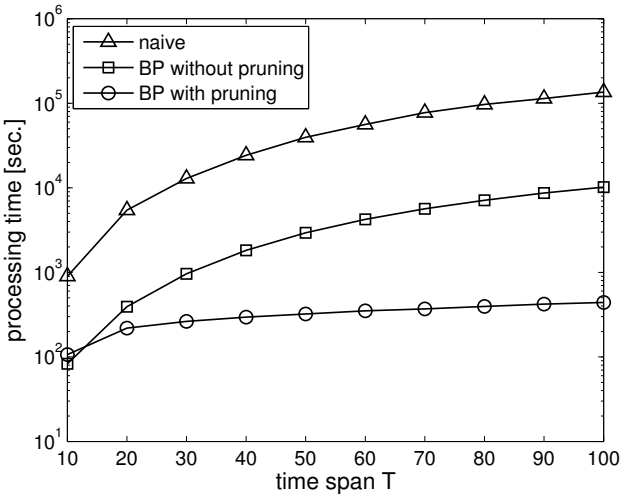


Figure 1: Results for the blog network.

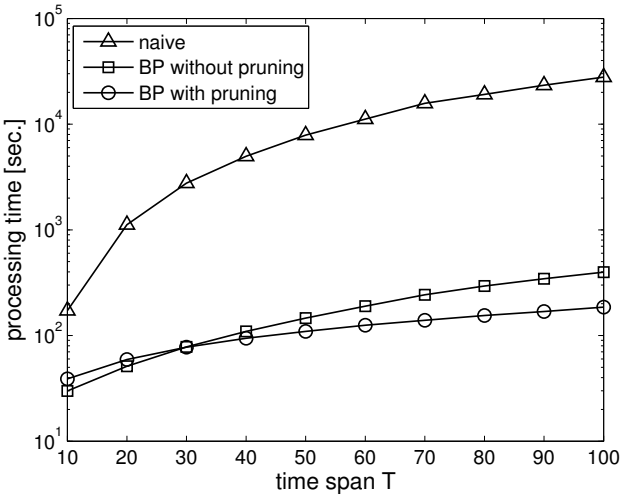


Figure 2: Results for the Wikipedia network.

span $T = 100$ is about 3 minutes, 6 minutes and 8 hours for the proposed method, the BP method without pruning and the naive method, respectively. Namely, the proposed method is about 2 and 150 times faster than the BP method without pruning and the naive method, respectively, for $T = 100$ in case of the Wikipedia network.

In general, the proposed method performs the best and the BP method without pruning follows with an exception that the proposed method can become slightly slower than the BP method without pruning in cases where T is small because of the overhead introduced in pruning. The two BP methods (with and without pruning) are much faster than the naive method. The performance difference between the proposed method and each of the other two methods increases as time-step (or time span) increases. Moreover, the same performance difference becomes larger for the blog network

than the Wikipedia network. The following simple analysis explains this. Consider the extreme case where $S(u,t) = S(v,t)$ for $\forall u, v \in A(t)$ and $d(w) = d$ for $\forall w \in S(v,t)$ ($v \in A(t)$) at some time-step t. We denote $|A(t)| = a$ and $|S(v,t)| = s$. Then, we have $\mathcal{N}_a = as$, $\mathcal{N}_b = sd$, $\mathcal{N}_{b'} = asd$ and $\mathcal{N}_c = asd'$ on average for time-step $t+1$ (see equations (1), (2) and (3)). Recall that d' is the expected number of the occupied links, which is calculated as pd, where p is the common diffusion probability for all links. Further assume that the pruning was ideal such that $\tilde{\mathcal{N}}_a = s$ and $\tilde{\mathcal{N}}_c = sd'$, which respectively denote the number of nodes identified in step 2-2a and the average number of links followed in step 2-2c" for the BP with pruning method. Then, if $ad' > d$, i.e., $ad'/d = ap > 1$ holds, the improvement ratios of the BP with pruning method over the naive method and the original BP method are respectively $asd/sd = a$ and $asd'/sd = ap$. From our experimental results, we can estimate a to be 310 for the blog network and 150 for the Wikipedia network. Then we obtain ap to be 31 and 1.5 respectively, which approximates the actual ratio each, 20 and 2.

5 Discussion

Here, we compare the method proposed in [Kimura *et al.*, 2007] that efficiently estimates the influence function also in the framework of bond percolation for the IC and the LT models. The same method is not applicable to the SIS model. The key idea there is to decompose the graph that is generated by the bond percolation into a set of strongly connected components (SCC) and efficiently calculate the node reachability. However, the layered graph in the proposed method is a directed acyclic tree and the SCC decomposition would not work effectively. The pruning technique in the proposed method is a new technique to improve the computational efficiency for the SIS model, just like the SCC decomposition is for the IC and the LT models.

In this paper we did not directly address the influential maximization problem, but only proposed a new method to efficiently estimate the influence function. We can think of two maximization problems, that is to find the initial active nodes with a specified number that maximize 1) the expected number of nodes that have been activated till the end of time-step T and 2) the expected number of active nodes at the end of time-step T. The proposed method can easily be extended to efficiently estimate the marginal gain of the objective function of each of the optimization problems when the problems are to be solved by greedy algorithms.

6 Conclusion

Finding influential nodes is one of the most central problems in the field of social network analysis. There are several models that simulate how various things, e.g., news, rumors, diseases, innovation, ideas, etc. diffuse across the network. One such realistic model is the *susceptible/infected/susceptible (SIS) model*, an information diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property, e.g., compared with the *susceptible/infected/recovered (SIR) model* where once activated nodes can never be deactivated/reactivated. We addressed the problem of efficiently estimating the influence function under the SIS model, i.e., estimating the expected number of activated nodes at time-step t for $t = 1, \cdots, T$ starting from an initially activated node v (for all $v \in V$) at time-step $t = 0$. We solved this problem by constructing a layered graph from the original social network by adding each layer on top of the existing layers as the time proceeds, and applying the bond percolation with a pruning strategy. We showed that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis. We further confirmed this by applying the proposed method to two real world networks taken from blog and Wikipedia data. Considerable reduction of computation time was achieved without degrading the accuracy.

References

[Adar and Adamic, 2005] E. Adar and L. A. Adamic. Tracking information epidemics in blogspace. In *WI'05*, pages 207–214, 2005.

[Agarwal and Liu, 2008] N. Agarwal and H. Liu. Blogosphere: Research issues, tools, and applications. *SIGKDD Explorations*, 10(1):18–31, 2008.

[Domingos and Richardson, 2001] P. Domingos and M. Richardson. Mining the network value of customers. In *KDD'01*, pages 57–66, 2001.

[Gruhl *et al.*, 2004] D. Gruhl, R. Guha, D. Liben-Nowell, and A. Tomkins. Information diffusion through blogspace. In *WWW'04*, pages 107–117, 2004.

[Kempe *et al.*, 2003] D. Kempe, J. Kleinberg, and E. Tardos. Maximizing the spread of influence through a social network. In *KDD'03*, pages 137–146, 2003.

[Kimura *et al.*, 2007] M. Kimura, K. Saito, and R. Nakano. Extracting influential nodes for information diffusion on a social network. In *AAAI'07*, pages 1371–1376, 2007.

[Kimura *et al.*, 2009] M. Kimura, K. Saito, and H. Motoda. Blocking links to minimize contamination spread in a social network. *ACM Transactions on Knowledge Discovery from Data*, 3(2):9:1–9:23, 2009.

[Leskovec *et al.*, 2007a] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. In *KDD'07*, pages 420–429, 2007.

[Leskovec *et al.*, 2007b] J. Leskovec, M. McGlohon, C. Faloutsos, , N. Glance, and M. Hurst. Patterns of cascading behavior in large blog graphs. In *SDM'07*, pages 551–556, 2007.

[Newman, 2003] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[Richardson and Domingos, 2002] M. Richardson and P. Domingos. Mining knowledge-sharing sites for viral marketing. In *KDD'02*, pages 61–70, 2002.

Discovering Influential Nodes for SIS models in Social Networks

Kazumi Saito[1], Masahiro Kimura[2], and Hiroshi Motoda[3]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu, Shiga 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of efficiently discovering the influential nodes in a social network under the *susceptible/infected/susceptible (SIS) model*, a diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property. We solve this problem by constructing a layered graph from the original social network with each layer added on top as the time proceeds, and applying the bond percolation with pruning and burnout strategies. We experimentally demonstrate that the proposed method gives much better solutions than the conventional methods that are solely based on the notion of centrality for social network analysis using two large-scale real-world networks (a blog network and a wikipedia network). We further show that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis and confirm this by experimentation. The properties of the influential nodes discovered are substantially different from those identified by the centrality-based heuristic methods.

1 Introduction

Social networks mediate the spread of various information including topics, ideas and even (computer) viruses. The proliferation of emails, blogs and social networking services (SNS) in the World Wide Web accelerates the creation of large social networks. Therefore, substantial attention has recently been directed to investigating information diffusion phenomena in social networks [1–3].

Overall, finding influential nodes is one of the most central problems in social network analysis. Thus, developing methods to do this on the basis of information diffusion is an important research issue. Widely-used fundamental probabilistic models of information diffusion are the *independent cascade (IC) model* and the *linear threshold (LT) model* [4, 5]. Researchers investigated the problem of finding a limited number of influential nodes that are effective for the spread of information under the above models [4,

6]. This combinatorial optimization problem is called the *influence maximization problem*. Kempe et al. [4] experimentally showed on large collaboration networks that the greedy algorithm can give a good approximate solution to this problem, and mathematically proved a performance guarantee of the greedy solution (i.e., the solution obtained by the greedy algorithm). Recently, methods based on bond percolation [6] and submodularity [7] were proposed for efficiently estimating the greedy solution. The influence maximization problem has applications in sociology and "viral marketing" [3], and was also investigated in a different setting (a descriptive probabilistic model of interaction) [8, 9]. The problem has recently been extended to influence control problems such as a contamination minimization problem [10].

The IC model can be identified with the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease [11, 5]. In the SIR model, only infected individuals can infect susceptible individuals, while recovered individuals can neither infect nor be infected. This implies that an individual is never infected with the disease multiple times. This property holds true for the LT model as well. However, there exist phenomena for which the property does not hold. For example, consider the following propagation phenomenon of a topic in the blogosphere: A blogger who has not yet posted a message about the topic is interested in the topic by reading the blog of a friend, and posts a message about it (i.e., becoming infected). Next, the same blogger reads a new message about the topic posted by some other friend, and may post a message (i.e., becoming infected) again. Most simply, this phenomenon can be modeled by an *susceptible/infected/susceptible (SIS) model* from the epidemiology. Like this example, there are many examples of information diffusion phenomena for which the SIS model is more appropriate, including the growth of hyper-link posts among bloggers [2], the spread of computer viruses without permanent virus-checking programs, and epidemic disease such as tuberculosis and gonorrhea [11].

We focus on an information diffusion process in a social network $G = (V, E)$ over a given time span T on the basis of an SIS model. Here, the SIS model is a stochastic process model, and the *influence* of a set of nodes H at time-step t, $\sigma(H, t)$, is defined as the expected number of infected nodes at time-step t when all the nodes in H are initially infected at time-step $t = 0$. We refer to σ as the *influence function* for the SIS model. Developing an effective method for estimating $\sigma(\{v\}, t)$, $(v \in V, t = 1, \ldots, T)$ is vital for various applications. Clearly, in order to extract influential nodes, we must estimate the value of $\sigma(\{v\}, t)$ for every node v and time-step t. Thus, we proposed a novel method based on the bond percolation with an effective pruning strategy to efficiently estimate $\{\sigma(\{v\}, t); v \in V, t = 1, \ldots, T\}$ for the SIS model in our previous work [12].

In this paper, we consider solving the influence maximization problems on a network $G = (V, E)$ under the SIS model. Here, unlike the cases of the IC and the LT models, we define two influence maximization problems, the *final-time maximization problem* and the *accumulated-time maximization problem*, for the SIS model. We introduce the greedy algorithm for solving the problems according to the work of Kempe et al. [4] for the IC and the LT models. Now, let us consider the problem of influence maximization at the final time step T (i.e., final-time maximization problem) as an example. We then note that for solving this problem by the greedy algorithm, we need a method for not only evaluating $\{\sigma(\{v\}, T); v \in V\}$, but also evaluating the *marginal influence*

gains $\{\sigma(H \cup \{v\}, T) - \sigma(H, T);\ v \in V \setminus H\}$ for any non-empty subset H of V. Needless to say, we can naively estimate the marginal influence gains for any non-empty subset H of V by simulating the SIS model[2]. However, this naive simulation method is overly inefficient and not practical at all. In this paper, by incorporating the new techniques (the pruning and the burnout methods) into the bond percolation method, we propose a method to efficiently estimate the marginal influence gains for any non-empty subset H of V, and apply it to approximately solve the two influence maximization problems for the SIS model by the greedy alogrithm. We show that the proposed method is expected to achieve a large reduction in computational cost by theoretically comparing computational complexity with other more naive methods. Further, using two large real networks, we experimentally demonstrate that the proposed method is much more efficient than the naive greedy method based on the bond percolation method. We also show that the discovered nodes by the proposed method are substantially different from and can result in considerable increase in the influence over the conventional methods that are based on the notion of various centrality measures.

2 Information Diffusion Model

Let $G = (V, E)$ be a directed network, where V and E ($\subset V \times V$) stand for the sets of all the nodes and (directed) links, respectively. For any $v \in V$, let $\Gamma(v; G)$ denote the set of the child nodes (directed neighbors) of v, that is,

$$\Gamma(v; G) = \{w \in V;\ (v, w) \in E\}.$$

2.1 SIS Model

An SIS model for the spread of a disease is based on the cycle of disease in a host. A person is first *susceptible* to the disease, and becomes *infected* with some probability when the person encounters an infected person. The infected person becomes susceptible to the disease soon without moving to the immune state. We consider a discrete-time SIS model for information diffusion on a network. In this context, infected nodes mean that they have just adopted the information, and we call these infected nodes *active* nodes.

We define the SIS model for information diffusion on G. In the model, the diffusion process unfolds in discrete time-steps $t \geq 0$, and it is assumed that the state of a node is either active or inactive. For every link $(u, v) \in E$, we specify a real value $p_{u,v}$ with $0 < p_{u,v} < 1$ in advance. Here, $p_{u,v}$ is referred to as the *propagation probability* through link (u, v). Given an initial set of active nodes X and a time span T, the diffusion process proceeds in the following way. Suppose that node u becomes active at time-step t ($< T$). Then, node u attempts to activate every $v \in \Gamma(u; G)$, and succeeds with probability $p_{u,v}$. If node u succeeds, then node v will become active at time-step $t + 1$. If multiple active nodes attempt to activate node v in time-step t, then their activation attempts are sequenced in an arbitrary order. On the other hand, node u will become or remain inactive at time-step $t + 1$ unless it is activated from an active node in time-step t. The process terminates if the current time-step reaches the time limit T.

[2] Note that the method we proposed in [12] does not perform simulation.

2.2 Influence Function

For the SIS model on G, we consider a diffusion sample from an initially activated node set $H \subset V$ over time span T. Let $S(H,t)$ denote the set of active nodes at time-step t. Note that $S(H,t)$ is a random subset of V and $S(H,0) = H$. Let $\sigma(H,t)$ denote the expected number of $|S(H,t)|$, where $|X|$ stands for the number of elements in a set X. We call $\sigma(H,t)$ the *influence* of node set H at time-step t. Note that σ is a function defined on $2^{|V|} \times \{0,1,\cdots,T\}$. We call the function σ the *influence function* for the SIS model over time span T on network G.

It is important to estimate the influence function σ efficiently. In theory we can simply estimate σ by the simulations based on the SIS model in the following way. First, a sufficiently large positive integer M is specified. For each $H \subset V$, the diffusion process of the SIS model is simulated from the initially activated node set H, and the number of active nodes at time-step t, $|S(H,t)|$, is calculated for every $t \in \{0,1,\cdots,T\}$. Then, $\sigma(H,t)$ is estimated as the empirical mean of $|S(H,t)|$'s that are obtained from M such simulations. However, this is extremely inefficient, and cannot be practical.

3 Influence Maximization Problem

We mathematically define the influence maximization problems on a network $G = (V,E)$ under the SIS model. Let K be a positive integer with $K < |V|$. First, we define the *final-time maximization problem*: Find a set H_K^* of K nodes to target for initial activation such that $\sigma(H_K^*;T) \geq \sigma(H;T)$ for any set H of k nodes, that is, find

$$H_K^* = \arg \max_{\{H \subset V;\, |H|=K\}} \sigma(H;T). \tag{1}$$

Second, we define the *accumulated-time maximization problem*: Find a set H_K^* of K nodes to target for initial activation such that $\sigma(H_K^*;1) + \cdots + \sigma(H_K^*;T) \geq \sigma(H;1) + \cdots + \sigma(H;T)$ for any set H of k nodes, that is, find

$$H_K^* = \arg \max_{\{H \subset V;\, |H|=K\}} \sum_{t=1}^{T} \sigma(H;t). \tag{2}$$

The first problem cares only how many nodes are influenced at the time of interest. For example, in an election campaign it is only those people who are convinced to vote the candidate at the time of voting that really matter and not those who were convinced during the campaign but changed their mind at the very end. Maximizing the number of people who actually vote falls in this category. The second problem cares how many nodes have been influenced throughout the period of interest. For example, maximizing the amount of product purchase during a sales campaign falls in this category.

4 Proposed Method

Kempe et al. [4] showed the effectiveness of the greedy algorithm for the influence maximization problem under the IC and LT models. In this section, we introduce the

greedy algorithm for the SIS model, and describe some techniques (the bond percolation method, the pruning method, and the burnout method) for efficiently solving the influence maximization problem under the greedy algorithm, together with some arguments for evaluating the computational complexity for these methods.

4.1 Greedy Algorithm

We approximately solve the influence maximization problem by the greedy algorithm. Below we describe this algorithm for the final-time maximization problem:

Greedy algorithm for the final-time maximization problem:
$\mathcal{A}$**1.** Set $H \leftarrow \emptyset$.
$\mathcal{A}$**2.** For $k = 1$ to K do the following steps:
$\mathcal{A}$**2-1.** Choose a node $v_k \in V \setminus H$ maximizing $\sigma(H \cup \{v\}, T)$.
$\mathcal{A}$**2-2.** Set $H \leftarrow H \cup \{v_k\}$.
$\mathcal{A}$**3.** Output H.

Here we can easily modify this algorithm for the accumulated-time maximization problem by replacing step $\mathcal{A}$2-1 as follows:

Greedy algorithm for the accumulated-time maximization problem:
$\mathcal{A}$**1.** Set $H \leftarrow \emptyset$.
$\mathcal{A}$**2.** For $k = 1$ to K do the following steps:
$\mathcal{A}$**2-1'.** Choose a node $v_k \in V \setminus H$ maximizing $\sum_{t=1}^{T} \sigma(H \cup \{v\}, t)$.
$\mathcal{A}$**2-2.** Set $H \leftarrow H \cup \{v_k\}$.
$\mathcal{A}$**3.** Output H.

Let H_K denote the set of K nodes obtained by this algorithm. We refer to H_K as the *greedy solution* of size K. Then, it is known that

$$\sigma(H_K, t) \ \geq \ \left(1 - \frac{1}{e}\right) \sigma(H_K^*, t),$$

that is, the quality guarantee of H_k is assured [4]. Here, H_k^* is the exact solution defined by Equation (1) or (2).

To implement the greedy algorithm, we need a method for estimating all the marginal influence degrees $\{\sigma(H \cup \{v\}, t);\ v \in V \setminus H\}$ of H in step $\mathcal{A}$2-1 or $\mathcal{A}$2-1' of the algorithm. In the subsequent subsections, we propose a method for efficiently estimating the influence function σ over time span T for the SIS model on network G.

4.2 Layered Graph

We build a layered graph $G^T = (V^T, E^T)$ from G in the following way. First, for each node $v \in V$ and each time-step $t \in \{0, 1, \cdots, T\}$, we generate a copy v_t of v at time-step t. Let V_t denote the set of copies of all $v \in V$ at time-step t. We define V^T by $V^T = V_0 \cup V_1 \cup \cdots \cup V_T$. In particular, we identify V with V_0. Next, for each link $(u, v) \in E$, we generate T links (u_{t-1}, v_t), $(t \in \{1, \cdots, T\})$, in the set of nodes V^T. We set $E_t = \{(u_{t-1}, v_t);\ (u, v) \in E\}$, and define E^T by $E^T = E_1 \cup \cdots \cup E_T$. Moreover, for any

link (u_{t-1}, v_t) of the layered graph G^T, we define the occupation probability q_{u_{t-1},v_t} by $q_{u_{t-1},v_t} = p_{u,v}$.

Then, we can easily prove that the SIS model with propagation probabilities $\{p_e; e \in E\}$ on G over time span T is equivalent to the *bond percolation process (BP) with occupation probabilities* $\{q_e; e \in E^T\}$ on G^T.[3] Here, the BP process with occupation probabilities $\{q_e; e \in E^T\}$ on G^T is the random process in which each link $e \in E^T$ is independently declared "occupied" with probability q_e. We perform the BP process on G^T, and generate a graph constructed by occupied links, $\tilde{G}^T = (V^T, \tilde{E}^T)$. Then, in terms of information diffusion by the SIS model on G, an occupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information propagates at time-step t, and an unoccupied link $(u_{t-1}, v_t) \in E_t$ represents a link $(u, v) \in E$ through which the information does not propagate at time-step t. For any $v \in V \setminus H$, let $F(H \cup \{v\}; \tilde{G}^T)$ be the set of all nodes that can be reached from $H \cup \{v\} \in V_0$ through a path on the graph $\tilde{G}^T$. When we consider a diffusion sample from an initial active node $v \in V$ for the SIS model on G, $F(H \cup \{v\}; \tilde{G}^T) \cap V_t$ represents the set of active nodes at time-step t, $S(H \cup \{v\}, t)$.

4.3 Bond Percolation Method

Using the equivalent BP process, we present a method for efficiently estimating influence function σ. We refer to this method as the *BP method*. Unlike the naive method, the BP method simultaneously estimates $\sigma(H \cup \{v\}, t)$ for all $v \in V \setminus H$. Moreover, the BP method does not fully perform the BP process, but performs it partially. Note first that all the paths from nodes $H \cup \{v\}$ $(v \in V \setminus H)$ on the graph $\tilde{G}^T$ represent a diffusion sample from the initial active nodes $H \cup \{v\}$ for the SIS model on G. Let L' be the set of the links in G^T that is not in the diffusion sample. For calculating $|S(H \cup \{v\}, t)|$, it is unnecessary to determine whether the links in L' are occupied or not. Therefore, the BP method performs the BP process for only an appropriate set of links in G^T. The BP method estimates σ by the following algorithm:

BP method:

$\mathcal{B}$**1.** Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

$\mathcal{B}$**2.** Repeat the following procedure M times:

$\mathcal{B}$**2-1.** Initialize $S(H \cup \{v\}, 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.

$\mathcal{B}$**2-2.** For $t = 1$ to T do the following steps:

$\mathcal{B}$**2-2a.** Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)$.

$\mathcal{B}$**2-2b.** Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

$\mathcal{B}$**2-2c.** For each $v \in A(t-1)$, compute $S(H \cup \{v\}, t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t)$, and set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t) + |S(H \cup \{v\}, t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$.

$\mathcal{B}$**3.** For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

[3] The SIS model over time span T on G can be exactly mapped onto the IC model on G^T [4]. Thus, the result follows from the equivalence of the BP process and the IC model [11, 4, 6].

Note that $A(t)$ finally becomes the set of information source nodes that have at least an active node at time-step t, that is, $A(t) = \{v \in V \setminus H;\, S(H \cup \{v\}, t) \neq \emptyset\}$. Note also that $B(t-1)$ is the set of nodes that are activated at time-step $t-1$ by some source nodes, that is, $B(t-1) = \bigcup_{v \in V} S(H \cup \{v\}, t-1)$.

Now we estimate the computational complexity of the BP method in terms of the number of the nodes, $\mathcal{N}_a$, that are identified in step $\mathcal{B}$2-2a, the number of the coin-flips, $\mathcal{N}_b$, for the BP process in step $\mathcal{B}$2-2b, and the number of the links, $\mathcal{N}_c$, that are followed in step $\mathcal{B}$2-2c. Let $d(v)$ be the number of out-links from node v (i.e., out-degree of v) and $d'(v)$ the average number of occupied out-links from node v after the BP process. Here we can estimate $d'(v)$ by $\sum_{w \in \Gamma(v;G)} p_{v,w}$. Then, for each time-step $t \in \{1, \cdots, T\}$, we have

$$\begin{aligned} \mathcal{N}_a &= \sum_{v \in A(t-1)} |S(H \cup \{v\}, t-1)|, \\ \mathcal{N}_b &= \sum_{w \in B(t-1)} d(w), \\ \mathcal{N}_c &= \sum_{v \in A(t-1)} \sum_{w \in S(H \cup \{v\}, t-1)} d'(w) \end{aligned} \tag{3}$$

on average.

In order to compare the computational complexity of the BP method to that of the naive method, we consider mapping the naive method onto the BP framework, that is, separating the coin-flip process and the link-following process. We can easily verify that the following algorithm in the BP framework is equivalent to the naive method:

A method that is equivalent to the naive method:

$\mathcal{B}$1. Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

$\mathcal{B}$2. Repeat the following procedure M times:

$\mathcal{B}$2-1. Initialize $S(H \cup \{v\}, 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$.

$\mathcal{B}$2-2. For $t = 1$ to T do the following steps:

$\mathcal{B}$2-2b'. For each $v \in A(t-1)$, perform the BP process for the links from $S(H \cup \{v\}, t-1)$ in G^T, and generate the graph $\tilde{G}_t(v)$ constructed by the occupied links.

$\mathcal{B}$2-2c'. For each $v \in A(t-1)$, compute $S(H \cup \{v\}; t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t(v))$, and set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t) + |S(H \cup \{v\}, t)|$ and $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$.

$\mathcal{B}$3. For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Then, for each $t \in \{1, \cdots, T\}$, the number of coin-flips, $\mathcal{N}_{b'}$, in step $\mathcal{B}$2-2b' is

$$\mathcal{N}_{b'} = \sum_{v \in A(t-1)} \sum_{w \in S(H \cup \{v\}, t-1)} d(w), \tag{4}$$

and the number of the links, $\mathcal{N}_{c'}$, followed in step $\mathcal{B}$2-2c' is equal to $\mathcal{N}_c$ in the BP method on average. From equations (3) and (4), we can see that $\mathcal{N}_{b'}$ is much larger than $\mathcal{N}_{c'} = \mathcal{N}_c$, especially for the case where the diffusion probabilities are small. We can also see that $\mathcal{N}_{b'}$ is generally much larger than each of $\mathcal{N}_a$ and $\mathcal{N}_b$ in the BP method for

a real social network. In fact, since such a network generally includes large clique-like subgraphs, there are many nodes $w \in V$ such that $d(w) \gg 1$, and we can expect that $\sum_{v \in A(t-1)} |S(H \cup \{v\}, t-1)| \gg |\bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)|$ $(= |B(t-1)|)$. Therefore, the BP method is expected to achieve a large reduction in computational cost.

4.4 Pruning Method

In order to further improve the computational efficiency of the BP method, we introduce a pruning technique and propose a method referred to as the *BP with pruning method*. The key idea of the pruning technique is to utilize the following property: Once we have $S(H \cup \{u\}, t_0) = S(H \cup \{v\}, t_0)$ at some time-step t_0 on the course of the BP process for a pair of information source nodes, u and v, then we have $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$ for all $t > t_0$. The BP with pruning method estimates σ by the following algorithm:

BP with pruning method:
$\mathcal{B}$1. Set $\sigma(H \cup \{v\}, t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.
$\mathcal{B}$2. Repeat the following procedure M times:
$\mathcal{B}$2-1". Initialize $S(H \cup \{v\}; 0) = H \cup \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V \setminus H$.
$\mathcal{B}$2-2. For $t = 1$ to T do the following steps:
$\mathcal{B}$2-2a. Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(H \cup \{v\}, t-1)$.
$\mathcal{B}$2-2b. Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.
$\mathcal{B}$2-2c". For each $v \in A(t-1)$, compute $S(H \cup \{v\}, t) = \bigcup_{w \in S(H \cup \{v\}, t-1)} \Gamma(w; \tilde{G}_t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(H \cup \{v\}, t) \neq \emptyset$, and set $\sigma(H \cup \{u\}, t) \leftarrow \sigma(H \cup \{u\}, t) + |S(H \cup \{v\}, t)|$ for each $u \in C(v)$.
$\mathcal{B}$2-2d. Check whether $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(H \cup \{u\}, t) = S(H \cup \{v\}, t)$.
$\mathcal{B}$3. For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\sigma(H \cup \{v\}, t) \leftarrow \sigma(H \cup \{v\}, t)/M$, and output $\sigma(H \cup \{v\}, t)$.

Basically, by introducing step $\mathcal{B}$2-2d and reducing the size of $A(t)$, the proposed method attempts to improve the computational efficiency in comparison to the original BP method. For the proposed method, it is important to implement efficiently the equivalence check process in step $\mathcal{B}$2-2d. In our implementation, we first classify each $v \in A(t)$ according to the value of $n = |S(H \cup \{v\}, t)|$, and then perform the equivalence check process only for those nodes with the same n value.

4.5 Burnout Method

In order to further improve the computational efficiency of the BP with pruning method, we additionally introduce a burnout technique and propose a method referred to as the *BP with pruning and burnout method*. More specifically, we focus on the fact that maximizing the marginal influence degree $\sigma(H \cup \{v\}, t)$ with respect to $v \in V \setminus H$ is equivalent to maximizing the marginal influence gain $\phi_H(v, t) = \sigma(H \cup \{v\}, t) - \sigma(H, t)$. Here on the course of the BP process for a newly added information source node v, maximizing $\phi_H(v, t)$ reduces to maximizing $|S(H \cup \{v\}, t) \setminus S(H, t)|$ on average. The BP with pruning and burnout method estimates ϕ_H by the following algorithm:

BP with pruning and burnout methods:

*C*1. Set $\phi_H(v,t) \leftarrow 0$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$.

*C*2. Repeat the following procedure M times:

*C*2-1. Initialize $S(H;0) = H$, and $S(\{v\};0) = \{v\}$ for each $v \in V \setminus H$, and set $A(0) \leftarrow V \setminus H$, $A(1) \leftarrow \emptyset, \cdots, A(T) \leftarrow \emptyset$, and $C(v) \leftarrow \{v\}$ for each $v \in V \setminus H$.

*C*2-2. For $t = 1$ to T do the following steps:

*C*2-2a. Compute $B(t-1) = \bigcup_{v \in A(t-1)} S(\{v\}, t-1) \cup S(H, t-1)$.

*C*2-2b. Perform the BP process for the links from $B(t-1)$ in G^T, and generate the graph $\tilde{G}_t$ constructed by the occupied links.

*C*2-2c. Compute $S(H,t) = \bigcup_{w \in S(H,t-1)} \Gamma(w; \tilde{G}_t)$, and for each $v \in A(t-1)$, compute $S(\{v\},t) = \bigcup_{w \in S(\{v\},t-1)} \Gamma(w; \tilde{G}_t) \setminus S(H,t)$, set $A(t) \leftarrow A(t) \cup \{v\}$ if $S(\{v\},t) \neq \emptyset$, and set $\phi_H(\{u\},t) \leftarrow \phi_H(\{u\},t) + |S(\{v\},t)|$ for each $u \in C(v)$.

*C*2-2d. Check whether $S(\{u\},t) = S(\{v\},t)$ for $u, v \in A(t)$, and set $C(v) \leftarrow C(v) \cup C(u)$ and $A(t) \leftarrow A(t) \setminus \{u\}$ if $S(\{u\},t) = S(\{v\},t)$.

*C*3. For each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$, set $\phi_H(\{v\},t) \leftarrow \phi_H(\{v\},t)/M$, and output $\phi_H(\{v\},t)$.

Intuitively, compared with the BP with pruning method, by using the burnout technique, we can substantially reduce the size of the active node set from $S(H \cup \{v\},t)$ to $S(\{v\},t)$ for each $v \in V \setminus H$ and $t \in \{1, \cdots, T\}$. Namely, in terms of computational costs described by Equation (3), we can expect to obtain smaller numbers for $\mathcal{N}_a$ and $\mathcal{N}_c$ when $H \neq \emptyset$. However, how effectively the proposed method works will depend on several conditions such as network structure, time span, values of diffusion probabilities, and so on. We will do a simple analysis later and experimentally show that it is indeed effective.

5 Experimental Evaluation

In the experiments, we report our evaluation results on the final-time maximization problem due to the space limitation.

5.1 Network Data and Settings

In our experiments, we employed two datasets of large real networks used in [10], which exhibit many of the key features of social networks.

The first one is a trackback network of Japanese blogs. The network data was collected by tracing the trackbacks from one blog in the site "goo (http://blog.goo.ne.jp/)" in May, 2005. We refer to the network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications by putting trackbacks on each other's blogs. The blog network had $12,047$ nodes and $79,920$ directed links.

The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. We refer to the network data as the Wikipedia network. The Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

For the SIS model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any link $(u, v) \in E$, that is, $p_{u,v} = p$. According to [4, 2], we set the value of p relatively small. In particular, we set the value of p to a value smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. We decided to set $p = 0.1$ for the blog network and $p = 0.01$ for the Wikipedia network. Also, for the time span T, we set $T = 30$.

For the bond percolation method, we need to specify the number M of performing the bond percolation process. According to [12], we set $M = 10,000$ for estimating influence degrees for the blog and Wikipedia networks.

All our experimentation was undertaken on a single PC with an Intel Dual Core Xeon X5272 3.4GHz processor, with 32GB of memory, running under Linux.

5.2 Comparison Methods

First, we compared the proposed method with three heuristics from social network analysis with respect to the solution quality. They are based on the notions of "degree centrality", "closeness centrality", and "betweenness centrality" that are commonly used as influence measure in sociology [13]. Here, the betweenness of node v is defined as the total number of shortest paths between pairs of nodes that pass through v, the closeness of node v is defined as the reciprocal of the average distance between v and other nodes in the network, and the degree of node v is defined as the number of links attached to v. Namely, we employed the methods of choosing nodes in decreasing order of these centralities. We refer to these methods as the *betweenness method*, the *closeness method*, and the *degree method*, respectively.

Next, to evaluate the effectiveness of the pruning and the burnout strategies, we compared the proposed method with the naive greedy method based on the BP method with respect to the processing time. Hereafter, we refer to the naive greedy method based on the BP method as the BP method for short.

5.3 Solution Quality Comparison

We first compared the quality of the solution H_K of the proposed method with that of the betweenness, the closeness, and the degree methods for solving the problem of the influence maximization at the final time step T. Clearly, the quality of H_K can be evaluated by the influence degree $\sigma(H_K, T)$. We estimated the value of $\sigma(H_K, T)$ by using the bond percolation method with $M = 10,000$ according to [12].

Figures 1 and 2 show the influence degree $\sigma(H_K, T)$ as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In the figures, the circles, triangles, diamonds, and squares indicate the results for the proposed, the betweenness, the closeness, and the degree methods, respectively. The proposed method performs the best for both networks, while the betweenness method follows for the blog dataset and the degree method follows for the Wikipedeia dataset. Note that how each of the conventional heuristics performs depends on the characteristics of the network structure. These results imply that the proposed method works effectively, and outperforms the conventional heuristics from social network analysis.

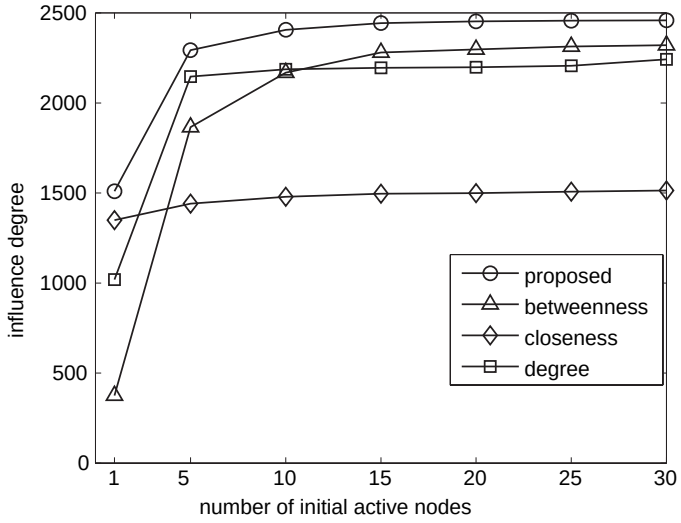


Fig. 1. Comparison of solution quality for the blog network.

It is interesting to note that the k nodes ($k = 1, 2, ..., K$) that are discovered to be most influential by the proposed method are substantially different from those that are found by the conventional centrality-based heuristic methods. For example, the best node ($k = 1$) chosen by the proposed method for the blog dataset is ranked 118 for the betweenness method, 659 for the closeness method and 6 for the degree method, and the 15th node ($k = 15$) by the proposed method is ranked 1373, 8848 and 507 for the corresponding conventional methods, respectively. The best node ($k = 1$) chosen by the proposed method for the Wikipedia dataset is ranked 580 for the betweenness method, 2766 for the closeness method and 15 for the degree method, and the 15th node ($k = 15$) by the proposed method is ranked 265, 2041, and 21 for the corresponding conventional methods, respectively. It is hard to find a correlation between these rankings, but for the smaller k, it appears that degree centrality measure is better than the other centrality measures, which can be inferred from Figures 1 and 2.

5.4 Processing Time Comparison

Next, we compared the processing time of the proposed method (BP with pruning and burnout method) with that of the BP method. Let $\tau(K, T)$ denote the processing time of a method for solving the problem of the influece maximization at the final time step T, where K is the number of initial active nodes. Figures 3 and 4 show the processing time difference $\Delta\tau(K, T) = \tau(K, T) - \tau(K - 1, T)$ as a function of the number of initial active nodes K for the blog and the Wikipedia networks, respectively. In these figures, the circles, and crosses indicate the results for the proposed and the BP methods, respectively. Note that $\Delta\tau(K, T)$ decreases as K increases for the proposed method, whereas $\Delta\tau(K, T)$ increases for the BP method. This means that the difference in the total processing time

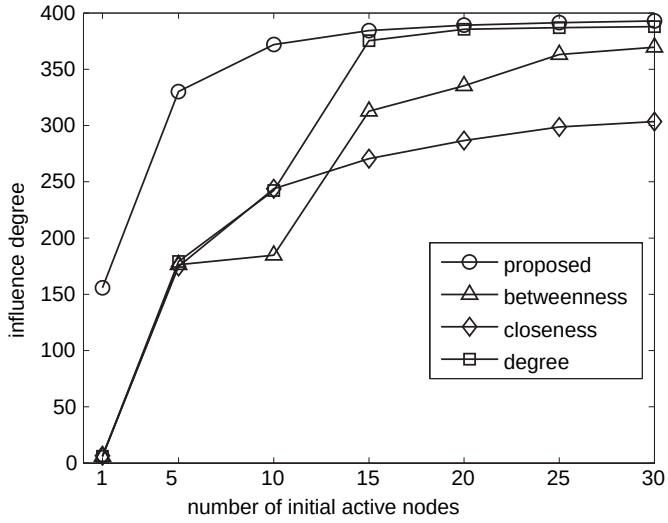


Fig. 2. Comparison of solution quality for the Wikipedia network.

becomes increasingly larger as K increases. In case of the blog dataset, the total processing time for $K = 5$ is about 2 hours for the proposed method and 100 hours for the BP methods. Namely, the proposed method is about 50 times faster than the BP method for $K = 5$. The same is true for the Wikipedia dataset. The total processing time for $K = 5$ is about 0.5 hours for the proposed method and 9 hours the BP methods, and the proposed method is about 18 times faster than the BP method for $K = 5$. These results confirm that the proposed method is much more efficient than the BP method, and can be practical.

6 Discussion

The influence function $\sigma(\cdot, T)$ is submodular [4]. For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. [7] have recently presented a lazy evaluation method that leads to far fewer (expensive) evaluations of the marginal increments $f(H \cup \{v\}) - f(H)$, $(v \in V \setminus H)$ in the greedy algorithm for $H \neq \emptyset$, and achieved an improvement in speed. Note here that their method requires evaluating $f(v)$ for all $v \in V$ at least. Thus, we can apply their method to the influence maximization problem for the SIS model, where the influence function $\sigma(\cdot, T)$ is evaluated by simulating the corresponding random process. It is clear that 1) this method is more efficient than the naive greedy method that does not employ the BP method and instead evaluates the influence degrees by simulating the diffusion phenomena, and 2) further the both methods become the same for $K = 1$ and empirically estimate the influence function $\sigma(\cdot, T)$ by probabilistic simulations. These methods also require M to be specified in advance as a parameter, where M is the number of simula-

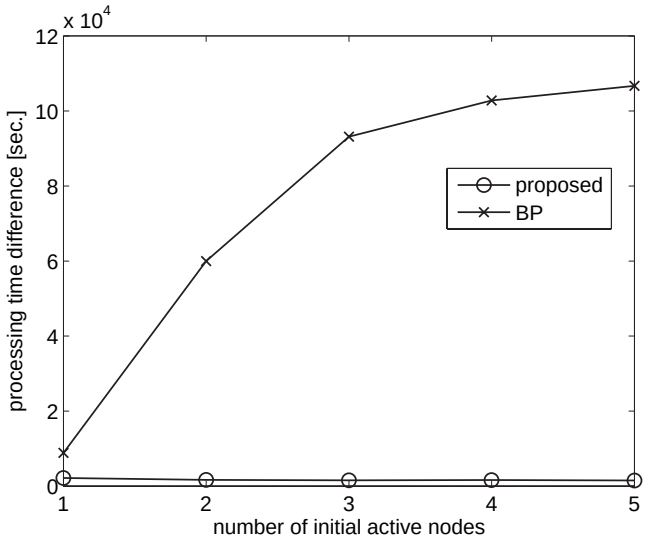


Fig. 3. Comparison of processing time for the blog network.

tions. Note that the BP and the simulation methods can estimate influence degree $\sigma(v, t)$ with the same accuracy by using the same value of M (see [12]). Moreover, as shown in [12], estimating influence function $\sigma(\cdot, 30)$ by $10,000$ simulations needed more than 35.8 hours for the blog dataset and 7.6 hours for the Wikipedia dataset, respectively. However, the proposed method for $K = 30$ needed less than 7.0 hours for the blog dataset and 3.2 hours for the Wikipedia dataset, respectively. Therefore, it is clear that the proposed method can be faster than the method by Leskovec [7] for the influence maximization problem for the SIS model.

7 Conclusion

Finding influential nodes is one of the most central problems in the field of social network analysis. There are several models that simulate how various things, e.g., news, rumors, diseases, innovation, ideas, etc. diffuse across the network. One such realistic model is the *susceptible/infected/susceptible (SIS) model*, an information diffusion model where nodes are allowed to be activated multiple times. The computational complexity drastically increases because of this multiple activation property, e.g., compared with the *susceptible/infected/recovered (SIR) model* where once activated nodes can never be deactivated/reactivated. We addressed the problem of efficiently discovering the influential nodes under the SIS model, i.e., estimating the expected number of activated nodes at time-step t for $t = 1, \cdots, T$ starting from an initially activated node set $H \in V$ at time-step $t = 0$. We solved this problem by constructing a layered graph from the original social network by adding each layer on top of the existing layers as the time proceeds, and applying the bond percolation with a pruning strategy. We showed

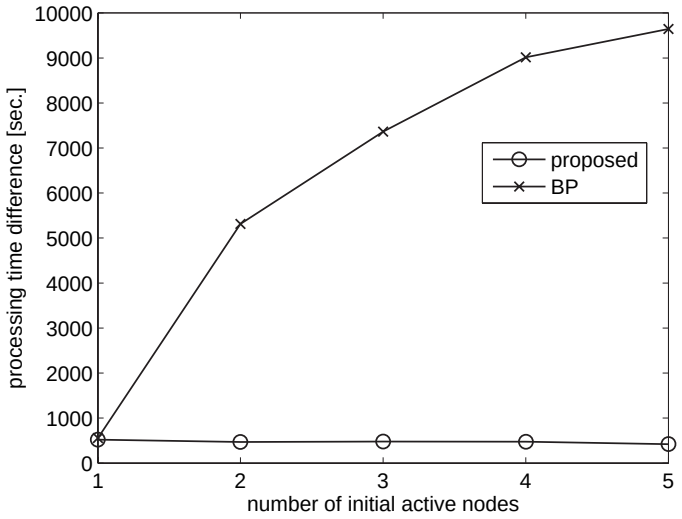


Fig. 4. Comparison of processing time for the Wikipedia network.

that the computational complexity of the proposed method is much smaller than the conventional naive probabilistic simulation method by a theoretical analysis. We applied the proposed method to two different types of influence maximization problem, i.e. discovering the K most influential nodes that together maximize the expected influence degree at the time of interest or the expected influence degree over the time span of interest. Both problems are solved by the greedy algorithm taking advantage of the submodularity of the objective function. We confirmed by applying to two real world networks taken from blog and Wikipedia data that the proposed method can achieve considerable reduction of computation time without degrading the accuracy compared with the naive simulation method, and discover nodes that are more influential than the nodes identified by the conventional methods based on the various centrality measures.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Adar, E., Adamic, L.A.: Tracking information epidemics in blogspace. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). (2005) 207–214

2. Leskovec, J., McGlohon, M., Faloutsos, C., Glance, N., Hurst, M.: Patterns of cascading behavior in large blog graphs. In: Proceedings of the 2007 SIAM International Conference on Data Mining (SDM'07). (2007) 551–556
3. Agarwal, N., Liu, H.: Blogosphere: Research issues, tools, and applications. SIGKDD Explorations **10** (2008) 18–31
4. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
5. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004). (2004) 107–117
6. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
7. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007). (2007) 420–429
8. Domingos, P., Richardson, M.: Mining the network value of customers. In: Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2001). (2001) 57–66
9. Richardson, M., Domingos, P.: Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002). (2002) 61–70
10. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
11. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
12. Kimura, M., Saito, K., Motoda, H.: Efficient estimation of influence functions fot sis model on social networks. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-09). (2009)
13. Wasserman, S., Faust, K.: Social network analysis. Cambridge University Press, Cambridge, UK (1994)

Learning Continuous-Time Information Diffusion Model for Social Behavioral Data Analysis

Kazumi Saito[1], Masahiro Kimura[2], Kouzou Ohara[3], and Hiroshi Motoda[4]

[1] School of Administration and Informatics, University of Shizuoka
52-1 Yada, Suruga-ku, Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp
[2] Department of Electronics and Informatics, Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp
[3] Department of Integrated Information Technology, Aoyama Gakuin Univesity
Kanagawa 229-8558, Japan
ohara@it.aoyama.ac.jp
[4] Institute of Scientific and Industrial Research, Osaka University
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

Abstract. We address the problem of estimating the parameters for a continuous time delay independent cascade (CTIC) model, a more realistic model for information diffusion in complex social network, from the observed information diffusion data. For this purpose we formulate the rigorous likelihood to obtain the observed data and propose an iterative method to obtain the parameters (time-delay and diffusion) by maximizing this likelihood. We apply this method first to the problem of ranking influential nodes using the network structure taken from two real world web datasets and show that the proposed method can predict the high ranked influential nodes much more accurately than the well studied conventional four heuristic methods, and second to the problem of evaluating how different topics propagate in different ways using a real world blog data and show that there are indeed differences in the propagation speed among different topics.

1 Introduction

The rise of the Internet and the World Wide Web accelerates the creation of various large-scale social networks, and considerable attention has been brought to social networks as an important medium for the spread of information [1–5]. Innovation, topics and even malicious rumors can propagate through social networks in the form of so-called "word-of-mouth" communications. This forms a virtual society forming various kinds of communities. Just like a real world society, some community grows rapidly and some other shrinks. Likewise, some information propagates quickly and some other only slowly. Good things remain and bad things diminish as if there is a natural selection. The social network offers a nice platform to study a mechanism of society dynamics and behavior of humans, each as a member of the society. In this paper, we

address the problem of how information diffuses through the social network, in particular how different topics propagate differently by inducing a diffusion model that can handle continuous time delay.

There are several models that simulate information diffusion through a network. A widely-used model is the *independent cascade (IC)*, a fundamental probabilistic model of information diffusion [6, 7], which can be regarded as the so-called *susceptible/infected/recovered (SIR) model* for the spread of a disease [2]. This model has been used to solve such problems as the *influence maximization problem* which is to find a limited number of nodes that are influential for the spread of information [7, 8] and the *influence minimization problem* which is to suppress the spread of undesirable information by blocking a limited number of links [9]. The IC model requires the parameters that represent diffusion probabilities through links to be specified in advance. Since the true values of the parameters are not available in practice, this poses yet another problem of estimating them from the observed data [10].

One of the drawbacks of the IC model is that it cannot handle time-delays for information propagation, and we need a model to explicitly represent time delay. Gruhl et al. is the first to extend the IC model to include the time-delay [3]. Their model now has the parameters that represent time-delays through links as well as the parameters that represent diffusion probabilities through links. They presented a method for estimating the parameter values from the observed data using an EM-like algorithm, and experimentally showed its effectiveness using sparse Erdös-Renyi networks. However, it is not clear what they are optimizing in deriving the update formulas of the parameter values. Further, they treated the time as a discrete variable, which means that it is assumed that information propagate in a synchronized way in a sense that each node can be activated only at a specific time. In reality, time flows continuously and thus information, too, propagates on this continuous time axis. For any node, information must be able to be received at any time from other nodes and must be allowed to propagate to yet other nodes at any other time, both in an asynchronous way. Thus, for a realistic behavior analyses of information diffusion, we need to adopt a model that explicitly represents continuous time delay.

In this paper, we deal with an information diffusion model that incorporates continuous time delay based on the IC model (referred to as CTIC model), and propose a novel method for estimating the values of the parameters in the model from a set of information diffusion results that are observed as time-sequences of infected (active) nodes. What makes this problem difficult is that incorporating time-delay makes the time-sequence observation data structural. There is no way of knowing from the data which node activated which other node that comes later in the sequence. We introduce an objective function that rigorously represents the likelihood of obtaining such observed data sequences under the CTIC model on a given network, and derive an iterative algorithm by which the objective function is maximized. First we test the convergence performance of the proposed method by applying it to the problem of ranking influential nodes using the network structure taken from two real world web datasets and show that the parameters converge to the correct values by the iterative procedure and can predict the high ranked influential nodes much more accurately than the well studied conventional four heuristic methods. Second we apply the method to the problem of be-

havioral analysis of topic propagation, i.e., evaluating how different topics propagate in different ways, using a real world blog data and show that there are indeed differences in the propagation speed among different topics.

2 Information Diffusion Model and Learning Problem

We first define the IC model according to [7], and then introduce the continuous-time IC model. After that, we formulate our learning problem.

We mathematically model the spread of information through a directed network $G = (V, E)$ without self-links, where V and E ($\subset V \times V$) stands for the sets of all the nodes and links, respectively. We call nodes *active* if they have been influenced with the information. In the model, it is assumed that nodes can switch their states only from inactive to active, but not from active to inactive. Given an initial set S of active nodes, we assume that the nodes in S have first become active at an initial time, and all the other nodes are inactive at that time.

In this paper, node u is called a *child node* of node v if $(v, u) \in E$, and node u is called a *parent node* of node v if $(u, v) \in E$. For each node $v \in V$, let $F(v)$ and $B(v)$ denote the set of child nodes of v and the set of parent nodes of v, respectively,

$$F(v) = \{w \in V;\ (v, w) \in E\}, \quad B(v) = \{u \in V;\ (u, v) \in E\}.$$

2.1 Independent Cascade Model

Let us describe the definition of the IC model. In this model, for each link (u, v), we specify a real value $\lambda_{u,v}$ with $0 < \lambda_{u,v} < 1$ in advance. Here $\lambda_{u,v}$ is referred to as the *diffusion probability* through link (u, v).

The diffusion process unfolds in discrete time-steps $t \geq 0$, and proceeds from a given initial active set S in the following way. When a node u becomes active at time-step t, it is given a single chance to activate each currently inactive child node v, and succeeds with probability $\lambda_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. If multiple parent nodes of v become active at time-step t, then their activation attempts are sequenced in an arbitrary order, but all performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

2.2 Continuous-Time Independent Cascade Model

Next, we extend the IC model so as to allow continuous-time delays, and refer to the extended model as the *continuous-time independent cascade (CTIC) model*.

In the CTIC model, for each link $(u, v) \in E$, we specify real values $r_{u,v}$ and $\kappa_{u,v}$ with $r_{u,v} > 0$ and $0 < \kappa_{u,v} < 1$ in advance. We refer to $r_{u,v}$ and $\kappa_{u,v}$ as the *time-delay parameter* and the *diffusion parameter* through link (u, v), respectively.

The diffusion process unfolds in continuous-time t, and proceeds from a given initial active set S in the following way. Suppose that a node u becomes active at time t. Then, node u is given a single chance to activate each currently inactive child node v. We

choose a delay-time δ from the exponential distribution with parameter $r_{u,v}$. If node v is not active before time $t + \delta$, then node u attempts to activate node v, and succeeds with probability $\kappa_{u,v}$. If u succeeds, then v will become active at time $t + \delta$. Under the continuous time framework, it is unlikely that multiple parent nodes of v attempt to activate v for the activation at time $t + \delta$. But if they do, their activation attempts are sequenced in an arbitrary order. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set S, let $\varphi(S)$ denote the number of active nodes at the end of the random process for the CTIC model. Note that $\varphi(S)$ is a random variable. Let $\sigma(S)$ denote the expected value of $\varphi(S)$. We call $\sigma(S)$ the *influence degree* of S for the CTIC model.

2.3 Learning problem

For the CTIC model on network G, we define the time-delay parameter vector $\boldsymbol{r}$ and the diffusion parameter vector $\boldsymbol{\kappa}$ by

$$\boldsymbol{r} = (r_{u,v})_{(u,v)\in E}, \quad \boldsymbol{\kappa} = (\kappa_{u,v})_{(u,v)\in E}.$$

In practice, the true values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ are not available. Thus, we must estimate them from past information diffusion histories observed as sets of active nodes.

We consider an observed data set of M independent information diffusion results,

$$\mathcal{D}_M = \{D_m;\ m = 1, \cdots, M\}.$$

Here, each D_m is a time-sequence of active nodes in the mth information diffusion result,

$$D_m = \langle D_m(t);\ t \in \mathcal{T}_m \rangle, \qquad \mathcal{T}_m = \langle t_m, \cdots, T_m \rangle,$$

where $D_m(t)$ is the set of all the nodes that have first become active at time t, and $\mathcal{T}_m$ is the observation-time list; t_m is the observed initial time and T_m is the observed final time. We assume that for any active node v in the mth information diffusion result, there exits some $t \in \mathcal{T}_m$ such that $v \in D_m(t)$. Let $t_{m,v}$ denote the time at which node v becomes active in the mth information diffusion result, i.e., $v \in D_m(t_{m,v})$. For any $t \in \mathcal{T}_m$, we set

$$C_m(t) = \bigcup_{\tau \in \mathcal{T}_m \cap \{s;\, s < t\}} D_m(\tau)$$

Note that $C_m(t)$ is the set of active nodes before time t in the mth information diffusion result. We also interpret D_m as referring to the set of all the active nodes in the mth information diffusion result for convenience sake. In this paper, we consider the problem of estimating the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$.

3 Proposed Method

We explain how we estimate the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ from $\mathcal{D}_M$. Here, we limit ourselves to outline the derivations of the proposed method due to the lack of space. We also briefly mention how we do behavioral analysis with the method.

3.1 Likelihood function

For the learning problem described above, we strictly derive the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ to use as our objective function.

First, we consider any node $v \in D_m$ with $t_{m,v} > 0$ for the mth information diffusion result. Let $\mathcal{A}_{m,u,v}$ denote the probability density that a node $u \in B(v) \cap C_m(t_{m,v})$ activates the node v at time $t_{m,v}$, that is,

$$\mathcal{A}_{m,u,v} = \kappa_{u,v} r_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})). \tag{1}$$

Let $\mathcal{B}_{m,u,v}$ denote the probability that the node v is not activated from a node $u \in B(v) \cap C_m(t_{m,v})$ within the time-period $[t_{m,u}, t_{m,v}]$, that is,

$$\begin{aligned} \mathcal{B}_{m,u,v} &= 1 - \kappa_{u,v} \int_{t_{m,u}}^{t_{m,v}} r_{u,v} \exp(-r_{u,v}(t - t_{m,u}))dt \\ &= \kappa_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) + (1 - \kappa_{u,v}). \end{aligned} \tag{2}$$

If there exist multiple active parents for the node v, i.e., $\eta = |B(v) \cap C_m(t_{m,v})| > 1$, we need to consider possibilities that each parent node succeeds in activating v at time $t_{m,v}$. However, in case of the continuous time delay model, we can ignore simultaneous activations by multiple active parents due to the continuous property. Thus, the probability density that the node v is activated at time $t_{m,v}$, denoted by $h_{m,v}$, can be expressed as

$$\begin{aligned} h_{m,v} &= \sum_{u \in B(v) \cap C_m(t_{m,v})} \mathcal{A}_{m,u,v} \left(\prod_{x \in B(v) \cap C_m(t_{m,v}) \setminus \{u\}} \mathcal{B}_{m,x,v} \right). \\ &= \prod_{x \in B(v) \cap C_m(t_{m,v})} \mathcal{B}_{m,x,v} \sum_{u \in B(v) \cap C_m(t_{m,v})} \mathcal{A}_{m,u,v} (\mathcal{B}_{m,u,v})^{-1}. \end{aligned} \tag{3}$$

Note that we are not able to know which node u actually activated the node v. This can be regarded as a hidden structure.

Next, for the mth information diffusion result, we consider any link $(v, w) \in E$ such that $v \in C_m(T_m)$ and $w \notin D_m$. Let $g_{m,v,w}$ denote the probability that the node w is not activated by the node v within the observed time period $[t_m, T_m]$. We can easily derive the following equation:

$$g_{m,v,w} = \kappa_{v,w} \exp(-r_{v,w}(T_m - t_{m,v})) + (1 - \kappa_{v,w}). \tag{4}$$

Here we can naturally assume that each information diffusion process finished sufficiently earlier than the observed final time, i.e., $T_m \gg \max\{t; D_m(t) \neq \emptyset\}$. Thus, as $T_m \to \infty$ in equation (4), we assume

$$g_{m,v,w} = 1 - \kappa_{v,w}. \tag{5}$$

Therefore, by using equations (3), (5), and the independence properties, we can define the likelihood function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ with respect to $\boldsymbol{r}$ and $\boldsymbol{\kappa}$ by

$$\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = \prod_{m=1}^{M} \left(\prod_{t \in \mathcal{T}_m} \prod_{v \in D_m(t)} h_{m,v} \prod_{v \in D_m} \prod_{w \in F(v) \setminus D_m} g_{m,v,w} \right). \tag{6}$$

Here, we retained the product with respect to $v \in D_m(t)$ for completeness, but in practice there is only one v in $D_m(t)$.

In this paper, we focus on the above situation (i.e., equation (5)) for simplicity, but we can easily modify our method to cope with the general one (i.e., equation (4)). Thus, our problem is to obtain the values of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, which maximize equation (6). For this estimation problem, we derive a method based on an iterative algorithm in order to stably obtain its solution.

3.2 Estimation method

We describe our estimation method. Let $\overline{\boldsymbol{r}} = (\overline{r}_{u,v})$ and $\overline{\boldsymbol{\kappa}} = (\overline{\kappa}_{u,v})$ be the current estimates of $\boldsymbol{r}$ and $\boldsymbol{\kappa}$, respectively. For each $v \in D_m$ and $u \in B(v) \cap C_m(t_{m,v})$, we define $\alpha_{m,u,v}$ by

$$\alpha_{m,u,v} \;=\; \mathcal{A}_{m,u,v}(\mathcal{B}_{m,u,v})^{-1} \,/ \sum_{x \in B(v) \cap C_m(t_{m,v})} \mathcal{A}_{m,x,v}(\mathcal{B}_{m,x,v})^{-1}. \tag{7}$$

Let $\overline{\mathcal{A}}_{m,u,v}$, $\overline{\mathcal{B}}_{m,u,v}$, $\overline{h}_{m,v}$, and $\overline{\alpha}_{m,u,v}$ denote the values of $\mathcal{A}_{m,u,v}$, $\mathcal{B}_{m,u,v}$, $h_{m,v}$, and $\alpha_{m,u,v}$ calculated by using $\overline{\boldsymbol{r}}$ and $\overline{\boldsymbol{\kappa}}$, respectively.

From equations (3), (5), (6), we can transform our objective function $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ as follows:

$$\log \mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M) = Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) - H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}), \tag{8}$$

where $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ is defined by

$$Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) = \sum_{m=1}^{M} \left(\sum_{t \in \mathcal{T}_m} \sum_{v \in D_m(t)} Q_{m,v} + \sum_{v \in D_m} \sum_{w \in F(v) \setminus D_m} \log(1 - \kappa_{v,w}) \right),$$

$$Q_{m,v} = \sum_{u \in B(v) \cap C_m(t_{m,v})} \log(\mathcal{B}_{m,u,v}) + \sum_{u \in B(v) \cap C_m(t_{m,v})} \overline{\alpha}_{m,u,v} \log\left(\mathcal{A}_{m,u,v}(\mathcal{B}_{m,u,v})^{-1}\right) \tag{9}$$

and $H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ is defined by

$$H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}}) = \sum_{m=1}^{M} \sum_{t \in \mathcal{T}_m} \sum_{v \in D_m(t)} \sum_{u \in B(v) \cap C_m(t_{m,v})} \overline{\alpha}_{m,u,v} \log \alpha_{m,u,v}. \tag{10}$$

Since $H(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ is maximized at $\boldsymbol{r} = \overline{\boldsymbol{r}}$ and $\boldsymbol{\kappa} = \overline{\boldsymbol{\kappa}}$ from equation (10), we can increase the value of $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\kappa}; \mathcal{D}_M)$ by maximizing $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$ (see equation (8)). Note here that although $\log \mathcal{A}_{m,u,v}$ is a linear combination of $\log \kappa_{u,v}$, $\log r_{u,v}$, and $r_{u,v}$, $\log \mathcal{B}_{m,u,v}$ cannot be written as such a linear combination (see equations (1), (2)). In order to cope with this problem of $\log \mathcal{B}_{m,u,v}$, we transform $\log \mathcal{B}_{m,u,v}$ in the same way as above, and define $\beta_{m,u,v}$ by

$$\beta_{m,u,v} = \kappa_{u,v} \exp(-r_{u,v}(t_{m,v} - t_{m,u})) \,/\, \mathcal{B}_{m,u,v}$$

Finally, as the solution which maximizes $Q(\boldsymbol{r}, \boldsymbol{\kappa}; \overline{\boldsymbol{r}}, \overline{\boldsymbol{\kappa}})$, we obtain the following update formulas of our estimation method:

$$r_{u,v} = \frac{\sum_{m \in \mathcal{M}^+_{u,v}} \overline{\alpha}_{m,u,v}}{\sum_{m \in \mathcal{M}^+_{u,v}} (\overline{\alpha}_{m,u,v} + (1 - \overline{\alpha}_{m,u,v})\overline{\beta}_{m,u,v})(t_{m,v} - t_{m,u})},$$

$$\kappa_{u,v} = \frac{1}{|\mathcal{M}^+_{u,v}| + |\mathcal{M}^-_{u,v}|} \sum_{m \in \mathcal{M}^+_{u,v}} (\overline{\alpha}_{m,u,v} + (1 - \overline{\alpha}_{m,u,v})\overline{\beta}_{m,u,v}),$$

where $\mathcal{M}^{+}_{u,v}$ and $\mathcal{M}^{-}_{u,v}$ are defined by

$$\mathcal{M}^{+}_{u,v} = \{m \in \{1, \cdots, M\};\ u, v \in D_m,\ v \in F(u),\ t_{m,u} < t_{m,v}\},$$
$$\mathcal{M}^{-}_{u,v} = \{m \in \{1, \cdots, M\};\ u \in D_m,\ v \notin D_m,\ v \in F(u)\}.$$

Note that we can regard our estimation method as a kind of the EM algorithm. It should be noted here that each time iteration proceeds the value of the likelihood function never decreases and the iterative algorithm is guaranteed to converge.

3.3 Behavioral analysis

Thus far, we assumed that the parameters (time-delay and diffusion) can vary with respect to links but remain the same irrespective of the topic of information diffused, following Gruhl et al. [3]. However, they may be sensitive to the topic.

Our method can cope with this by assigning m to a topic, and placing a constraint that the parameters depends only on topics but not on links throughout the network G, that is $r_{m,u,v} = r_m$ and $\kappa_{m,u,v} = \kappa_m$ for any link $(u, v) \in E$. This constraint is required because, without this, we have only one piece of observation for each (m, u, v) and there is no way to learn the parameters. Noting that we can naturally assume that people behave quite similarly for the same topic, this constraint should be acceptable. Under this setting, we can easily obtain the parameter update formulas. Using each pair of the estimated parameters, (r_m, κ_m), we can analyze the behavior of people with respect to the topics of information, by simply plotting (r_m, κ_m) as a point of 2-dimensional space.

3.4 Simple case analysis

Here, we analyze a few basic properties of the proposed method under simple settings. Assume that a node v became active at time t after receiving certain information. We denote the active parent nodes of v by $u_1, \cdots, u_N$. First, we consider a simple case that diffusion parameter κ is 1 for all links, time-delay parameter r is a constant and the same for all links, and the activation times of $u_1, \cdots, u_N$ are all zeros. Then, as is given in equation (3), the probability density that the node v is activated at time t by one of the parent nodes, can be expressed as follows

$$h_v = \sum_{n=1}^{N} r \exp(-rt) \left(1 - \int_0^t r \exp(-r\tau) d\tau\right)^{N-1} = Nr \exp(-Nrt).$$

Similarly, for the case that the parent nodes $u_1, \cdots, u_N$ became active at times $t_1, \cdots t_N$ $(< t)$, respectively, we easily obtain the following probability.

$$h_v = Nr \exp\left(-Nr\left(t - \frac{1}{N}\sum_{n=1}^{N} t_n\right)\right).$$

The maximum likelihood is attained by maximizing $\log h_v$ with respect to r, and the average delay time is obtained as follows:

$$r^{-1} = N\left(t - \frac{1}{N}\sum_{n=1}^{N} t_n\right).$$

We can see that this estimation is N times larger than the simple average of time differences. In other words, the information diffuses more quickly when there exist multiple active parents, i.e., r^{-1}/N, and this fact matches our intuition. Thus simple statistics such as the average delay time may fail to provide the intrinsic property of information diffusion phenomena, and this suggests that an adequate information diffusion model is vital.

Next, we consider another simple case that the diffusion parameter κ and the time-delay parameter r are both uniform and constant for all links, and the activation times of $u_1, \cdots, u_N$ are all zeros. Here both parameters are variables. Then the probability density that the node v is activated at time t can be expressed as follows

$$h_v = N\kappa r \exp(-rt)(\kappa \exp(-rt) + (1-\kappa))^{N-1}.$$

Now, we consider maximizing $f(\kappa, r) = \log h_v$ with respect to κ and r. The first- and second-order derivatives of $f(\kappa, r)$ with respect to κ are given by

$$\frac{\partial f(\kappa, r)}{\partial \kappa} = \frac{1}{\kappa} + (N-1)\frac{\exp(-rt) - 1}{\kappa \exp(-rt) + (1-\kappa)}$$
$$\frac{\partial^2 f(\kappa, r)}{\partial \kappa \partial \kappa} = -\frac{1}{\kappa^2} - (N-1)\left(\frac{\exp(-rt) - 1}{\kappa \exp(-rt) + (1-\kappa)}\right)^2.$$

Since the above second-order derivative is negative definite for a given parameter r, we note that there exists a unique global solution to κ. The corresponding derivatives with respect to r are given by

$$\frac{\partial f(\kappa, r)}{\partial r} = \frac{1}{r} - t - (N-1)\frac{t\kappa \exp(-rt)}{\kappa \exp(-rt) + (1-\kappa)}$$
$$\frac{\partial^2 f(\kappa, r)}{\partial r \partial r} = -\frac{1}{r^2} + (N-1)\frac{t^2\kappa(1-\kappa)\exp(-rt)}{(\kappa \exp(-rt) + (1-\kappa))^2}.$$

Unfortunately, we cannot guarantee that the above second-order derivative is negative definite. However, most likely, this value is negative when $r \ll 1$, and can be positive when $r \gg 1$ in which case the shape of the objective function can be complex. We can speculate that the convergence is better for a smaller value of r. Later, in our experiments, we empirically evaluate this point by using the method described in 3.1 and 3.2 with $r = 2$ and $r = 0.5$, which are in the range that is widely explored by many existing studies. Clearly, we need to perform further theoretical and empirical studies because we are simultaneously estimating both diffusion and time-delay parameters, κ and r. However, the experiments show that our method is stable for the range of parameters we used, indicating that the likelihood function has favorable mathematical properties.

4 Experiments with Artificial data

We evaluated the effectiveness of the proposed learning method using the topologies of two large real network data. First, we evaluated how accurately it can estimate the parameters of the CTIC model from $\mathcal{D}_M$. Next, we considered applying our learning method to the problem of extracting influential nodes, and evaluated how well our learned model can predict the high ranked influential nodes with respect to influence degree $\sigma(v)$, $(v \in V)$ for the true CTIC model.

4.1 Experimental Settings

In our experiments, we employed two datasets of large real networks used in [9], which exhibit many of the key features of social networks. The first one is a trackback network of Japanese blogs. The network data was collected by tracing the trackbacks from one blog in the site *goo*[2] in May, 2005. We refer to this network data as the blog network. The blog network was a strongly-connected bidirectional network, where a link created by a trackback was regarded as a bidirectional link since blog authors establish mutual communications by putting trackbacks on each other's blogs. The blog network had $12,047$ nodes and $79,920$ directed links. The second one is a network of people that was derived from the "list of people" within Japanese Wikipedia. We refer to this network data as the Wikipedia network. The Wikipedia network was also a strongly-connected bidirectional network, and had $9,481$ nodes and $245,044$ directed links.

Here, we assumed the simplest case where $r_{u,v}$ and $\kappa_{u,v}$ are uniform throughout the network G, that is, $r_{u,v} = r$, $\kappa_{u,v} = \kappa$ for any link $(u, v) \in E$. One reason behind this assumption is that we can make fair comparison with the existing heuristics that are solely based on network structure (see 4.2). Another reason is that there is no need to acquire observation sequence data that at least pass through every link once. This drastically reduces the amount of data to learn the parameters. Then, our task is to estimate the values of r and κ. According to [7], we set the value of κ relatively small. In particular, we set the value of κ to a value smaller than $1/\bar{d}$, where $\bar{d}$ is the mean out-degree of a network. Since the values of $\bar{d}$ were about 6.63 and 25.85 for the blog and the Wikipedia networks, respectively, the corresponding values of $1/\bar{d}$ were about 0.15 and 0.03. Thus, as for the true value of the diffusion parameter κ, we decided to set $\kappa = 0.1$ for the blog network and $\kappa = 0.01$ for the Wikipedia network. As for the true value of the time-delay parameter r, we decided to investigate two cases: one with a relatively high value $r = 2$ (a short time-delay case) and the other with a relatively low value $r = 0.5$ (a long time-delay case) in both networks. We used the training data $\mathcal{D}_M$ in the learning stage, which is constructed by generating each D_m from a randomly selected initial active node $D_m(0)$ using the true CTIC model. T_m was chosen to be effectively ∞.

We note that the influence degree $\sigma(v)$ of a node v is invariant with respect to the values of the delay-parameter $\boldsymbol{r}$. In fact, the effect of $\boldsymbol{r}$ is to delay the timings when nodes become active, that is, parameter $r_{u,v}$ only controls how soon or late node v actually becomes active when node u activates node v. Therefore, nodes that can be activated are in indeed activated eventually after a sufficiently long time has elapsed, which is the case here, i.e. $T_m = \infty$. Thus, we can evaluate the $\sigma(v)$ of the CTIC model by the influence degree of v for the corresponding IC model. We estimated the influence degrees $\{\sigma(v); v \in V\}$ using the method of [8] with the parameter value $10,000$, where the parameter represents the number of bond percolation processes (we do not describe the method here due to the page limit). The average value and the standard deviation of the influence degrees was 87.5 and 131 for the blog network, and 8.14 and 18.4 for the Wikipedia network.

[2] `http://blog.goo.ne.jp/`

Table 1: Learning performance by the proposed method.

Blog network ($r = 2$)		
M	$\mathcal{E}_r$	$\mathcal{E}_\kappa$
20	0.013	0.015
40	0.010	0.010
60	0.008	0.008
80	0.007	0.007
100	0.005	0.005

Wikipedia network ($r = 2$)		
M	$\mathcal{E}_r$	$\mathcal{E}_\kappa$
20	0.036	0.034
40	0.024	0.016
60	0.013	0.015
80	0.012	0.013
100	0.006	0.011

Blog network ($r = 0.5$)		
M	$\mathcal{E}_r$	$\mathcal{E}_\kappa$
20	0.011	0.012
40	0.010	0.007
60	0.009	0.005
80	0.004	0.004
100	0.004	0.004

Wikipedia network ($r = 0.5$)		
M	$\mathcal{E}_r$	$\mathcal{E}_\kappa$
20	0.026	0.028
40	0.021	0.023
60	0.018	0.021
80	0.014	0.012
100	0.007	0.006

4.2 Comparison Methods

We compared the predicted result of the high ranked influential nodes for the true CTIC model by the proposed method with four heuristics widely used in social network analysis.

The first three of these heuristics are "degree centrality", "closeness centrality", and "betweenness centrality". These are commonly used as influence measure in sociology [11], where the out-degree of node v is defined as the number of links going out from v, the closeness of node v is defined as the reciprocal of the average distance between v and other nodes in the network, and the betweenness of node v is defined as the total number of shortest paths between pairs of nodes that pass through v. The fourth is "authoritativeness" obtained by the "PageRank" method [12]. We considered this measure since this is a well known method for identifying authoritative or influential pages in a hyperlink network of web pages. This method has a parameter ε; when we view it as a model of a random web surfer, ε corresponds to the probability with which a surfer jumps to a page picked uniformly at random [13]. In our experiments, we used a typical setting of $\varepsilon = 0.15$.

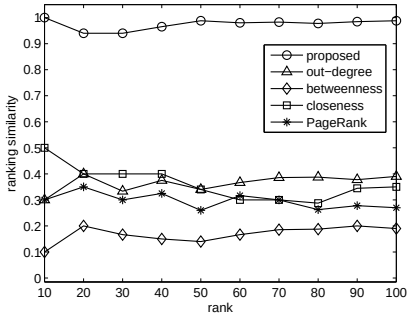
4.3 Experimental Results

First, we examined the parameter estimation accuracy by the proposed method. Let r_0 and κ_0 be the true values of the parameters r and κ, respectively, and let $\hat{r}$ and $\hat{\kappa}$ be the values of r and κ estimated by the proposed method, respectively. We evaluated the learning performance in terms of the error rates,

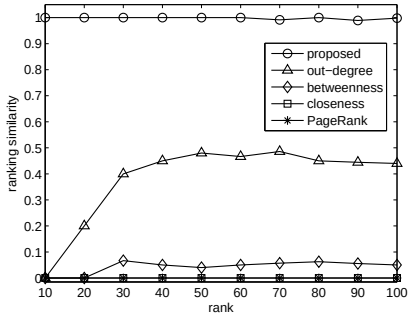
$$\mathcal{E}_r = \frac{|r_0 - \hat{r}|}{r_0}, \quad \mathcal{E}_\kappa = \frac{|\kappa_0 - \hat{\kappa}|}{\kappa_0}.$$

Table 1 shows the average values of $\mathcal{E}_r$ and $\mathcal{E}_\kappa$ for different numbers of training samples, M. For each M we repeated the same experiment 5 times independently, and for each experiment we tried 5 different initial values of the parameters that are randomly drawn from [0,1] with uniform distribution. The convergence criterion is

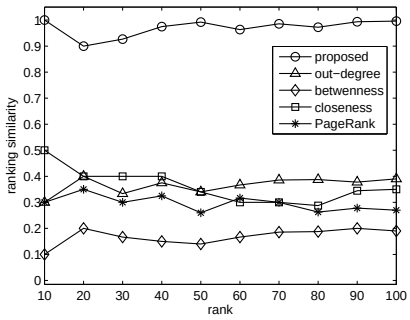
$$|\kappa^{(n)} - \kappa^{(n+1)}| + |r^{(n)} - r^{(n+1)}| < 10^{-12},$$



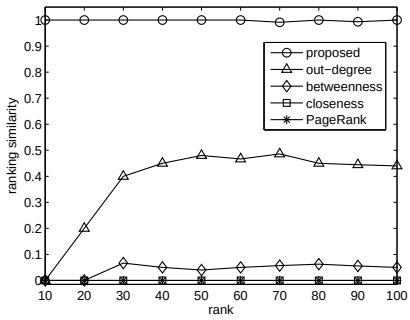
(a) blog network ($r = 2$)



(b) Wikipedia network ($r = 2$)



(c) blog network ($r = 0.5$)



(d) Wikipedia network ($r = 0.5$)

Fig. 1: Performance comparison in extracting influential nodes.

where the superscript (n) indicates the value for the nth iteration. Our algorithm converged at around 40 iterations for the blog data and 70 iterations for the Wikipedia data. Further, it is observed, as predicted by the simple case analysis in 3.4, that the convergence was faster for a smaller value of r. The converged values are close to the true values when there is a reasonable amount of training data. The results demonstrate the effectiveness of the proposed method.

Next, we compared the proposed method with the out-degree, the betweenness, the closeness, and the PageRank methods in terms of the capability of ranking the influential nodes. For any positive integer k ($\leq |V|$), let $L_0(k)$ be the true set of top k nodes, and let $L(k)$ be the set of top k nodes for a given ranking method. We evaluated the performance of the ranking method by the *ranking similarity* $F(k)$ at rank k, where $F(k)$ is defined by

$$F(k) = \frac{|L_0(k) \cap L(k)|}{k}.$$

We focused on ranking similarities only at high ranks since we are interested in extracting influential nodes. Figures 1a and 1c show the results for the blog network, and Figures 1b and 1d show the results for the Wikipedia network, where the true value of

r is $r = 2$ and $r = 0.5$ for Figures 1a and 1b, and Figures 1c and 1d, respectively. In these figures, circles, triangles, diamonds, squares, and asterisks indicate ranking similarity $F(k)$ as a function of rank k for the proposed, the out-degree, the betweenness, the closeness, and the PageRank methods, respectively. For the proposed method, we plotted the average value of $F(k)$ at k for 5 experimental results stated earlier in the case of $M = 100$. The proposed method gives far better results than the other heuristic based methods for the both networks, demonstrating the effectiveness of our proposed learning method.

5 Behavioral Analysis of Real World Blog Data

We applied our method to behavioral analysis using a real world blog data based on the method described in 3.3 and investigated how each topic spreads throughout the network.

5.1 Experimental Settings

The network we used is a real blogroll network in which bloggers are connected to each other. We note that when there is a blogroll link from blogger y to another blogger x, this means that y is a reader of the blog of x. Thus, we can assume that topics propagate from blogger x to blogger y. According to [14], we suppose that a topic is represented as a URL which can be tracked down from blog to blog. We used the database of a blog-hosting service in Japan called *Doblog* [3]. The database is constructed by all the Doblog data from October 2003 to June 2005, and contains $52,525$ bloggers and $115,552$ blogroll links.

We identified all the URLs mentioned in blog posts in the Doblog database, and constructed the following list for each URL from all the blog posts that contain the URL:

$$\langle (v_1, t_1), \cdots, (v_k, t_k) \rangle, \quad (t_1 < \cdots < t_k),$$

where v_i is a blogger who mentioned the URL in her/his blog post published at time t_i. By taking into account the blogroll relations for the list, we estimated such paths that the URL might propagate through the blogroll network. We extracted $7,356$ URL propagation paths from the Doblog dataset, where we ignored the URLs that only one blogger mentioned. Out of these, only those that are longer than 10 time steps are chosen for analyses, resulting into 172 sequences. Each sequence data represents a topic, and a topic can be distributed in multiple URLs. The same URL can appear in different sequences. Here note that the time stamp of each blog article is different from each other and thus, the time intervals in the sequence $< t_1, t_2, ..., t_k >$ are not a fixed constant.

5.2 Experimental Results

We ran the experiments for each identified URL and obtained the corresponding parameters κ and r. Figure 2 is a plot of the results for the major URLs. The horizontal axis

[3] Doblog(`http://www.doblog.com/`), provided by NTT Data Corp. and Hotto Link, Inc.

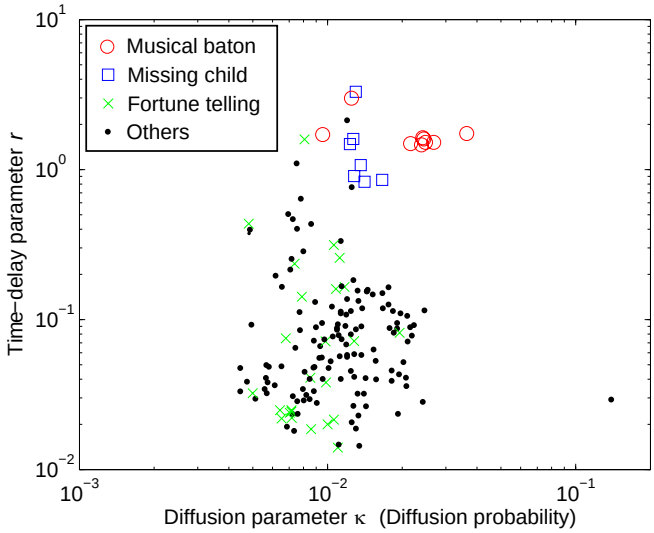


Fig. 2: Results for the Doblog database.

is the diffusion parameter κ and the vertical axis is the delay parameter r. The latter is normalized such that $r = 1$ corresponds to a delay of one day, meaning $r = 0.1$ corresponds delay of 10 days. We only explain three URLs that exhibit some interesting propagation properties. The circle is a URL that corresponds to the musical baton which is a kind of telephone game on the Internet. It has the following rules. First, a blogger is requested to respond to five questions about music by some other bologger (receive the baton) and the requested blogger replies to the questions and designate the next five bloggers with the same questions (pass the baton). It is shown that this kind of message propagates quickly (less than one day on the average) with a good chance (one out of 25 to 100 persons responds). This is probably because people are interested in this kind of message passing. The square is a URL that corresponds to articles about a missing child. This also propagates quickly with a meaningful probability (one out of 80 persons responds). This is understandable considering the urgency of the message. The cross is a URL that corresponds to articles about fortune telling. Peoples responses are diverse. Some responds quickly (less than one day) and some late (more than one month after), and they are more or less uniformly distributed. The diffusion probability is also nearly uniformly distributed. This reflects that each individual's interest is different on this topic. The dot is a URL that corresponds to one of the other topics. Interestingly, the one in the bottom right which is isolated from the rest is a post of an invitation to a rock music festival. This one has a very large probability of being propagated but with very large time delay. In general, it can be said that the proposed method can extract characteristic properties of a certain topics reasonably well only from the observation data.

6 Discussion

Being able to handle the time more precisely brings a merit to the analysis of such information diffusion as in a blog data because the time stamp is available in the unit of second. There are subtle cases where it is not self evident to which value to assign the time when the discretization has to be made. We have solved this problem.

There are many pieces of work in which time sequence data is analyzed assuming a certain model behind. Ours also falls in this category. The proposed approach brings in a new perspective in which it allows to use the structure of a complex network as a kind of background knowledge in a more refined way. There are also many pieces of work on topic propagation analyses, but they focus mostly on the analyses of average propagation speed (propagation speed distribution) and average life time. Our method is new and different in that we explicitly address the diffusion phenomena incorporating diffusion probability and time delay as well as the structure of the network.

The proposed method derives the learning algorithm in a principled way. The objective function has a clear meaning of the likelihood by which to obtain the observed data, and the parameter is iteratively updated in such a way to maximize the likelihood, guaranteeing the convergence. Due to the property of continuous time, we excluded the possibility that a node is activated simultaneously by multiple parent nodes. It is also straightforward to formulate the likelihood taking the possibility of the simultaneous activation into account. However, the numerical experiments revealed that the results are not as accurate as the current model. Having to explore millions of paths with very small probability does harm numerical computation. This is, in a sense, similar to the problem of feature selection in building a classifier. It is known that the existence of irrelevant features is harmful even though the classification algorithm can in theory ignore those irrelevant features.

The CTIC model is a continuous-time information diffusion model that extends the discrete-time model by Gruhl et al [15]. We note that their model is based on the popular IC model and they model the time-delay by a geometic distribition. In the CTIC model, we model a time-delay by an exponential distribution. Song et al [16] also modeled time-delays of information flow by exponential distributions in formulating an information flow model by a continuous-time Markov chain (i.e., a random-surfer model). Thus, we can regard the CTIC model as a natural extension to continuous-time information diffusion model based on the IC model, and investigating its characteristics can be an important research issue. As explained in Section 2.2, the CTIC model is rather complicated, and developing a learning algorithm of the CTIC model is challenging. In this paper, we presented an effective method for estimating the parameters of the CTIC model from observed data, and applied it to node-ranking and social behavioral data analysis. To the best of our knowledge, we are the first to formulate a continuous-time information diffusion model based on the IC model and a rigorous learning algorithm to estimate the model parameters from observation. We are not claiming that the model is most accurate. The time-delay distribution for real information diffusion must be more complex, and a power-law distribution and the like might be more suitable. Our future work includes incorpolating various more realistic distributions as the time-delay distribution.

The learning algorithm we proposed is a one-time batch processing. In reality the observation data are keep coming and the environment may change over time. It is not straightforward to convert the algorithm to incremental mode. The simplest way to cope with this is to use a fixed time window to collect data and use the parameters at the previous window as the initial guesses.

We consider that our proposed ranking method presents a novel concept of centrality based on the information diffusion model, i.e., *the CTIC model*. Actually, nodes identified as higher ranked by our method are substantially different from those by each of the conventional methods. This means that our method enables a new type of social network analysis if past information diffusion data are available. Note that this is not to claim to replace them with the proposed method, but simply to propose that it is an addition to them which has a different merit in terms of information diffusion.

We note that the analysis we showed in this paper is the simplest case where κ and r take a single value each for all the links in E. However, the method is very general. In a more realistic setting we can divide E into subsets $E_1, E_2, ..., E_N$ and assign a different value κ_n and r_n for all the links in each E_n. For example, we may divide the nodes into two groups: those that strongly influence others and those not, or we may divide the nodes into another two groups: those that are easily influenced by others and those not. We can further divide the nodes into multiple groups. If there is some background knowledge about the node grouping, our method can make the best use of it.

7 Conclusion

We emphasized the importance of incorporating continuous time delay for the behavioral analysis of information diffusion through a social network, and addressed the problem of estimating the parameters for a continuous time delay independent cascade (CTIC) model from the observed data by rigorously formulating the likelihood of obtaining these data and maximizing the likelihood iteratively with respect to the parameters (time-delay and diffusion). We tested the convergence performance of the proposed method by applying it to the problem of ranking influential nodes using the network structure from two real world web datasets and showed that the parameters converge to the correct values efficiently by the iterative procedure and can predict the high ranked influential nodes much more accurately than the well studied four heuristic methods. We further applied the method to the problem of behavioral analysis of topic propagation using a real world blog data and showed that there are indeed sensible differences in the propagation patterns in terms of delay and diffusion among different topics.

Acknowledgments

This work was partly supported by Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027, and JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147).

References

1. Newman, M.E.J., Forrest, S., Balthrop, J.: Email networks and the spread of computer viruses. Physical Review E **66** (2002) 035101
2. Newman, M.E.J.: The structure and function of complex networks. SIAM Review **45** (2003) 167–256
3. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. SIGKDD Explorations **6** (2004) 43–52
4. Domingos, P.: Mining social networks for viral marketing. IEEE Intelligent Systems **20** (2005) 80–82
5. Leskovec, J., Adamic, L.A., Huberman, B.A.: The dynamics of viral marketing. In: Proceedings of the 7th ACM Conference on Electronic Commerce (EC'06). (2006) 228–237
6. Goldenberg, J., Libai, B., Muller, E.: Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters **12** (2001) 211–223
7. Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003). (2003) 137–146
8. Kimura, M., Saito, K., Nakano, R.: Extracting influential nodes for information diffusion on a social network. In: Proceedings of the 22nd AAAI Conference on Artificial Intelligence (AAAI-07). (2007) 1371–1376
9. Kimura, M., Saito, K., Motoda, H.: Blocking links to minimize contamination spread in a social network. ACM Transactions on Knowledge Discovery from Data **3** (2009) 9:1–9:23
10. Saito, K., Kimura, M., Nakano, R., Motoda, H.: Finding influential nodes in a social network from information diffusion data. In: Proceedings of the International Workshop on Social Computing and Behavioral Modeling (SBP09). (2009) 138–145
11. Wasserman, S., Faust, K.: Social network analysis. Cambridge University Press, Cambridge, UK (1994)
12. Brin, S., L.Page: The anatomy of a large-scale hypertextual web search engine. Computer Networks and ISDN Systems **30** (1998) 107–117
13. Ng, A.Y., Zheng, A.X., Jordan, M.I.: Link analysis, eigenvectors and stability. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-01). (2001) 903–910
14. Adar, E., Adamic, L.A.: Tracking information epidemics in blogspace. In: Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence (WI'05). (2005) 207–214
15. Gruhl, D., Guha, R., Liben-Nowell, D., Tomkins, A.: Information diffusion through blogspace. In: Proceedings of the 13th International World Wide Web Conference (WWW 2004). (2004) 107–117
16. Song, X., Chi, Y., Hino, K., Tseng, B.L.: Information flow modeling based on diffusion rate for prediction and ranking. In: Proceedings of the 16th International World Wide Web Conference (WWW 2007). (2007) 191–200

Extracting Influential Nodes on a Social Network for Information Diffusion

Masahiro Kimura, Kazumi Saito, Ryohei Nakano, and Hiroshi Motoda

Abstract

We address the combinatorial optimization problem of finding the most influential nodes on a large-scale social network for two widely-used fundamental stochastic diffusion models. The past study showed that a greedy strategy can give a good approximate solution to the problem. However, a conventional greedy method faces a computational problem. We propose a method of efficiently finding a good approximate solution to the problem under the greedy algorithm on the basis of bond percolation and graph theory, and compare the proposed method with the conventional method in terms of computational complexity in order to theoretically evaluate its effectiveness. The results show that the proposed method is expected to achieve a great reduction in computational cost. We further experimentally demonstrate that the proposed method is much more efficient than the conventional method using large-scale real-world networks including blog networks.

Keywords

Social network analysis, Information diffusion model, Influence maximization problem, Bond percolation

Authors' Addresses:

Masahiro Kimura
Department of Electronics and Informatics
Ryukoku University
Otsu 520-2194, Japan
kimura@rins.ryukoku.ac.jp

Kazumi Saito
School of Administration and Informatics
University of Shizuoka
Shizuoka 422-8526, Japan
k-saito@u-shizuoka-ken.ac.jp

Ryohei Nakano
Department of Computer Science
Chubu University
Aichi 487-8501, Japan
nakano@cs.chubu.ac.jp

Hiroshi Motoda
Institute of Scientific and Industrial Research
Osaka University
Osaka 567-0047, Japan
motoda@ar.sanken.osaka-u.ac.jp

1 Introduction

The rise of the Internet and the World Wide Web has enabled us to investigate large-scale social networks, and there has been growing interest in social network analysis (Newman, 2001; McCallum et al., 2005; Leskovec et al., 2006). Here, a social network is the network of relationships and interactions among social entities such as individuals, groups of individuals, and organizations. Examples include blog networks, collaboration networks, and email networks.

The social network of interactions within a group of individuals plays a fundamental role in the spread of information, ideas, and innovations. In fact, a piece of information, such as the URL of a website that provides a new valuable service, can spread from one individual to another through the social network in the form of "word-of-mouth" communication. For example, the information of free email services such as Microsoft's Hotmail and Google's Gmail could spread largely through email networks. Thus, when we plan to market a new product, promote an innovation, or spread a new topic among a group of individuals, we can exploit social network effects. Namely, we can *target* a small number of influential individuals (e.g., giving free samples of the product, demonstrating the innovation, or offering the topic), and trigger a cascade of influence by which friends will recommend the product, promote the innovation, or propagate the topic to other friends. In this way, we can spread decisions in adopting the product, the innovation, or the topic through the social network from a small set of initial adopters to many individuals. Therefore, given a social network represented by a directed graph, a positive integer k, and a probabilistic model for the process by which a certain information spreads through the network, it is an important research issue in terms of sociology and *viral marketing* to find such a target set A_k^* of k nodes that maximizes the expected number of adopters of the information if A_k^* initially adopts it (Domingos and Richardson, 2001; Richardson and Domingos, 2002; Kempe et al, 2003; Kempe et al., 2005). Here, the expected number of nodes influenced by a target set is referred to as its *influence degree*, and this combinatorial optimization problem is called the *influence maximization problem* of size k.

Kempe et al. (2003) studied the influence maximization problem for two widely-used fundamental information diffusion models, the *independent cascade (IC) model* (Goldenberg, 2001; Kempe et al., 2003; Gruhl et al., 2004) and the *linear threshold (LT) model* (Watts, 2002; Kempe et al., 2003). They experimentally showed on large collaboration networks that for the influence maximization problem under the IC and LT models, the greedy algorithm significantly outperforms the high-degree and centrality heuristics that are commonly used in the sociology literature. Here, the high-degree heuristic chooses nodes in order of decreasing degrees, and the centrality heuristic chooses nodes in order of increasing average distance to other nodes in the

network. Moreover, they mathematically proved a performance guarantee of the greedy algorithm under these information diffusion models (i.e., the IC and LT models) by using an analysis framework based on submodular functions.

For the influence maximization problem of size k, the greedy algorithm iteratively finds a target set A_k of k nodes from the target set A_{k-1} of $k-1$ nodes that it has already found. Thus, it requires a method of computing all the *marginal influence degrees* of a given set A of nodes in the network. Here, for any node v that does not belong to A, the influence degree of target set $A \cup \{v\}$ is referred to as the *marginal influence degree* of A at v. However, it is an open question to compute influence degrees exactly by an efficient method, and therefore, the conventional method had to obtain good estimates for influence degrees by simulating the random process of the information diffusion model (i.e., the IC or LT model) many times (Kempe et al., 2003). Solving the influence maximization problem under the greedy algorithm needed a large amount of computation for large-scale networks.

In this paper, for the IC and LT models, we propose a method of efficiently estimating all the marginal influence degrees of a given set of nodes on the basis of bond percolation and graph theory, and apply it to approximately solving the influence maximization problem under the greedy algorithm. In order to theoretically evaluate the effectiveness of the proposed method for solving the influence maximization problem, we compare the proposed method with the conventional method in terms of computational complexity, and show that the proposed method is expected to achieve a large reduction in computational cost. Further, using large-scale real networks including blog networks, we experimentally demonstrate that the proposed method is much more efficient than the conventional method. Finally, we discuss some related work, and describe the conclusion.

2 Definitions

We examine the influence maximization problem on a network represented by a directed graph $G = (V, E)$ for the IC and LT models. Here, V and E are the sets of all the nodes and links in the network, respectively. Let N and L be the numbers of elements of V and E, respectively.

We first recall some basic notions from graph theory. Next, we define the IC and LT models on G according to the work of Kempe et al. (2003). Last, we give a mathematical definition of the influence maximization problem.

2.1 Graphs

We consider a directed graph $G = (V, E)$. If there is a directed link (u, v) from node u to node v, node v is called a *child node* of node u and node u is called a *parent node* of node v. For any $v \in V$, let $\Gamma(v)$ denote the set of all

the parent nodes of v. For a subset V' of V, graph $G' = (V', E')$ is called the *induced graph* of G to V' if $E' = E \cap (V' \times V')$.

We call $(u_0, \cdots, u_\ell)$ a *path* from node u_0 to node u_ℓ if we have $(u_{i-1}, u_i) \in E$, $(i = 1, \cdots, \ell)$. We say that node u can *reach* node v or node v is *reachable* from node u if there is a path from node u to node v. For a node v of the graph G, we define $F(v; G)$ to be the set of all the nodes that are reachable from v, and define $B(v; G)$ to be the set of all the nodes that can reach v. For any $A \subset V$, we set

$$F(A; G) = \bigcup_{v \in A} F(v; G), \ B(A; G) = \bigcup_{v \in A} B(v; G).$$

A *strongly connected component (SCC)* of G is a maximal subset C of V such that for all u, $v \in C$ there is a path from u to v. For a node v of G, we define $SCC(v; G)$ to be the SCC that contains v.

2.2 Information Diffusion Models

We consider mathematically modeling the spread of certain information through a social network $G = (V, E)$. In the IC and LT models, the following assumptions are made:

- A node is called *active* if it has adopted the information.
- The state of a node is either *active* or *inactive*.
- Nodes can switch from being inactive to being active, but cannot switch from being active to being inactive.
- The spread of the information through the network G is represented as the spread of active nodes on G.
- Given an initial set A of active nodes, we suppose that the nodes in A first become active and all the other nodes remain inactive at time-step 0.
- The diffusion process of active nodes unfolds in discrete time-steps $t \geq 0$.

2.2.1 Independent Cascade Model

First, we define the *independent cascade (IC) model*. In this model, we specify a real value $p_{u,v} \in [0, 1]$ for each directed link (u, v) in advance. Here, $p_{u,v}$ is referred to as the *propagation probability* through link (u, v). When an initial set A of active nodes is given, the diffusion process of active nodes proceeds according to the following randomized rule. When node u first becomes active at time-step t, it is given a single chance to activate

each of its currently inactive child nodes v, and succeeds with probability $p_{u,v}$. If u succeeds, then v will become active at time-step $t+1$. Here, if v has multiple parent nodes that become active at time-step t for the first time, then their activation attempts are sequenced in an arbitrary order, but performed at time-step t. Whether or not u succeeds, it cannot make any further attempts to activate v in subsequent rounds. The process terminates if no more activations are possible.

For an initial active set A ($\subset V$), let $\varphi(A)$ denote the number of active nodes at the end of the random process for the IC model. Note that $\varphi(A)$ is a random variable. Let $\sigma(A)$ denote the expected value of $\varphi(A)$. We call $\sigma(A)$ the *influence degree* of A.

2.2.2 Linear Threshold Model

Next, we define the *linear threshold (LT) model*. In this model, for any node $v \in V$, we in advance specify a *weight* $w_{u,v}$ (> 0) from its parent node u such that

$$\sum_{u \in \Gamma(v)} w_{u,v} \leq 1.$$

When an initial set A of active nodes is given, the diffusion process of active nodes proceeds according to the following randomized rule. First, for any node $v \in V$, a *threshold* θ_v is chosen uniformly at random from the interval $[0, 1]$. At time-step t, an inactive node v is influenced by each of its active parent nodes u according to weight $w_{u,v}$. If the total weight from active parent nodes of v is at least threshold θ_v, that is,

$$\sum_{u \in \Gamma_t(v)} w_{u,v} \geq \theta_v,$$

then v will become active at time-step $t+1$. Here, $\Gamma_t(v)$ stands for the set of parent nodes of v that are active at time-step t. The process terminates if no more activations are possible.

Note that the threshold θ_v models the tendency of node v to adopt the information when its parent nodes do. Note also that the LT model is a probabilistic model associated with the uniform distribution on $[0, 1]^N$. Further note that in the LT model it is the node thresholds that are random, while in the IC model it is the propagations through links that are random. Suppose that A is an initial set of active nodes. We define a random variable $\varphi(A)$ by the number of active nodes at the end of the random process for the LT model. Let $\sigma(A)$ denote the expected value of $\varphi(A)$. We call $\sigma(A)$ the *influence degree* of A. Note that these notations are the same as those for the IC model.

2.3 Influence Maximization Problem

We mathematically define the influence maximization problem on a network $G = (V, E)$ under the IC and LT models. Let k be a positive integer with $k < N$.

The *influence maximization problem* on G of size k is defined as follows: Find a set A_k^* of k nodes to target for initial activation such that $\sigma(A_k^*) \geq \sigma(S)$ for any set S of k nodes, that is, find

$$A_k^* \ = \ \mathrm{argmax}_{A \in \{S \subset V;\, |S|=k\}} \ \sigma(A), \tag{1}$$

where $|S|$ stands for the number of elements of set S.

3 Conventional Method

Kempe et al. (2003) showed the effectiveness of the greedy algorithm for the influence maximization problem under the IC and LT models. In this section, we introduce the greedy algorithm, and describe the conventional method for solving the influence maximization problem under the greedy algorithm. We, then, consider evaluating the computational complexity for the conventional method.

3.1 Greedy Algorithm

We approximately solve the influence maximization problem by the following greedy algorithm:

(G1) Set $A \leftarrow \emptyset$.

(G2) **for** $i = 1$ to k **do**

(G3) Choose a node $v_i \in V$ maximizing $\sigma(A \cup \{v\})$, $(v \in V \setminus A)$.

(G4) Set $A \leftarrow A \cup \{v_i\}$.

(G5) **end for**

Let A_k denote the set of k nodes obtained by this algorithm. We refer to A_k as the *greedy solution* of size k. Then, it is known that

$$\sigma(A_k) \ \geq \ \left(1 - \frac{1}{e}\right) \sigma(A_k^*),$$

that is, the quality guarantee of A_k is assured (Kempe et al., 2003). Here, A_k^* is the exact solution defined by Equation (1).

To implement the greedy algorithm, we need a method for estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the algorithm.

3.2 Conventional Method for Estimating Marginal Influence Degrees

For Step (G3) of the greedy algorithm, the conventional method estimated all the marginal influence degrees $\{\sigma(A\cup\{v\});\, v \in V\backslash A\}$ of A in the following way (Kempe et al., 2003): First, a sufficiently large positive integer M is specified. For any $v \in V \setminus A$, the random process of the diffusion model (IC or LT model) is run from the initial active set $A \cup \{v\}$, and the number $\varphi(A \cup \{v\})$ of final active nodes is counted. Each $\sigma(A \cup \{v\})$ is estimated as the empirical mean obtained from M such simulations.

Namely, the conventional method independently estimated $\sigma(A \cup \{v\})$ for all $v \in V \setminus A$ as follows:

1. **for** $m = 1$ to M **do**
2. Compute $\varphi(A \cup \{v\})$.
3. Set $x_m \leftarrow \varphi(A \cup \{v\})$.
4. **end for**
5. Set $\sigma(A \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_m$.

Here, each $\varphi(A \cup \{v\})$ is computed as follows:

1. Set $H_0 \leftarrow A \cup \{v\}$.
2. Set $t \leftarrow 0$.
3. **while** $H_t \neq \emptyset$ **do**
4. Set $H_{t+1} \leftarrow \{$the activated nodes at time $t + 1\}$.
5. Set $t \leftarrow t + 1$.
6. **end while**
7. Set $\varphi(A \cup \{v\}) \leftarrow \sum_{j=0}^{t-1} |H_j|$

3.3 Computational Complexity of Conventional Method

We consider evaluating the computational complexity of solving the influence maximization problem. For this purpose, we introduce the notion of *examined nodes*. Here, an *examined node* is a node that is actually visited by tracing incoming or outgoing links on the graph in question for the method when all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A are estimated in Step (G3) of the greedy algorithm. In Section 4.4, we describe the reason why we investigate the examined nodes for evaluating the computational complexity.

The computational complexity of the conventional method is evaluated in terms of the expected number of examined nodes. In order to estimate $\sigma(A \cup \{v\})$, $(v \in V \setminus A)$, it is necessary for any $v \in V \setminus A$ to simulate M times the random process of the information diffusion model (IC or LT model) from the initial active set $A \cup \{v\}$ on graph G. For each simulation, the set of examined nodes are the same as the set of active nodes in the process. Thus, we can estimate that the expected number $\mathcal{C}_0$ of examined nodes for the conventional method is

$$\mathcal{C}_0 = M \sum_{v \in V \setminus A} \sigma(A \cup \{v\}). \tag{2}$$

4 Proposed Method

We propose a method for efficiently estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm on the basis of bond percolation and graph theory, and evaluate the computational complexity, and compare it with that of the conventional method.

4.1 Bond Percolation

The IC and LT models are identified with *bond percolation models* which are defined below, and all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A are efficiently estimated by exploiting graph theoretic methods.

A *bond percolation* process on graph $G = (V, E)$ is the process in which each link of G is randomly designated either "occupied " or "unoccupied" according to some probability distribution. Here, in terms of information diffusion on a social network, occupied links represent the links through which the information propagates, and unoccupied links represent the links through which the information does not propagate. Let us consider the following set of L-dimensional vectors,

$$R_G = \left\{ r = (r_{u,v})_{(u,v) \in E} \in \{0,1\}^L \right\},$$

where L is the number of links in G. A bond percolation process on G is determined by a probability distribution $q(r)$ on R_G. Namely, for a random vector $r \in R_G$ drawn from $q(r)$, each link $(u, v) \in E$ is designated "occupied" if $r_{u,v} = 1$, and it is designated "unoccupied" if $r_{u,v} = 0$. Let E_r denote the set of all the occupied links for $r \in R_G$, and let G_r denote the graph (V, E_r). For each $r \in R_G$, we can consider the deterministic diffusion model $\mathcal{M}_r$ on G_r such that $F(A; G_r)$ becomes the final set of active nodes when A is an initial set of active nodes, where $F(A; G_r)$ is the set that is reachable from A on G_r (see, Section 2.1). By associating the diffusion model $\mathcal{M}_r$ on G_r with a probability distribution $q(r)$ on R_G, we define a stochastic diffusion model on G. We call this diffusion model the *bond percolation model* on G, and

refer to the probability distribution $q(r)$ on R_G as the *occupation probability distribution* of the bond percolation model.

We easily see that the IC model on G can be identified with the so-called *susceptible/infective/recovered (SIR) model* (Newman, 2003) for the spread of a disease on G, where the nodes that become active at time t in the IC model correspond to the infective nodes at time t in the SIR model. We recall that in the SIR model, an individual occupies one of the three states, "susceptible", "infected" and 'recovered", where a susceptible individual becomes infected with a certain probability when s/he is encountered an infected patient and subsequently recovers at a certain rate (see, Newman, 2003; Watts and Dodds, 2007). It is known that the SIR model on a network can be exactly mapped onto a bond percolation model on the same network (Grassberger, 1983; Newman, 2002; Kempe et al., 2003; Newman, 2003). Hence, we see that the IC model on G is equivalent to a bond percolation model on G, that is, these two models have the same probability distribution for the final set of active nodes given a target set. Here, for the IC model on G, the occupation probability distribution $q(r)$ of the corresponding bond percolation model is given by

$$q(r) = \prod_{(u,v)\in E} \left\{ (p_{u,v})^{r_{u,v}} (1-p_{u,v})^{1-r_{u,v}} \right\}, \quad (r \in R_G),$$

that is, each link (u,v) of G is independently declared to be "occupied" with probability $p_{u,v}$, where $p_{u,v}$ is the propagation probability through link (u,v) in the IC model.

On the other hand, Kempe et al. (2003) proved that the LT model on G can also be equivalent to a bond percolation model on G to derive the result that the influence degree function $\sigma(A)$ is submodular in the LT model. Here, for the LT model on G, the corresponding occupation probability distribution $q(r)$ is generated by declaring "occupied" and "unoccupied" links in the following way: For any $v \in V$, we pick at most one of the incoming links to v by selecting link (u,v) with probability $w_{u,v}$ and selecting no link with probability $1 - \sum_{u\in\Gamma(v)} w_{u,v}$. After this process, the picked links are declared to be "occupied" and the other links are declared to be "unoccupied". Here, $w_{u,v}$ is the weight of link (u,v) in the LT model. Specifically, $q(r)$ is described as follows:

$$q(r) \;=\; \prod_{v\in V} \prod_{u\in\Gamma(v)} \left\{ (w_{u,v})^{r_{u,v}} \left(1 - \sum_{u\in\Gamma(v)} w_{u,v} \right)^{\left(1-\sum_{u\in\Gamma(v)} r_{u,v}\right)} \right\},$$

where if $\sum_{u\in\Gamma(v)} w_{u,v} < 1$, $\sum_{u\in\Gamma(v)} r_{u,v} \le 1$ and if $\sum_{u\in\Gamma(v)} w_{u,v} = 1$, $\sum_{u\in\Gamma(v)} r_{u,v} = 1$.

4.2 Proposed Method for Estimating Marginal Influence Degrees

We present a method of estimating all the marginal influence degrees $\{\sigma(A\cup\{v\});\ v\in V\setminus A\}$ of A in Step (G3) of the greedy algorithm. As shown in the preceding section, the IC and LT models on G can be identified with the bond percolation models on G. Therefore, we have

$$\sigma(A\cup\{v\})=\sum_{r\in R_G}q(r)\,|F(A\cup\{v\};G_r)|$$

for any $v\in V\setminus A$, where $q(r)$ is the corresponding occupation probability distribution, and $F(A\cup\{v\};G_r)$ stands for the set of all the nodes that are reachable from $A\cup\{v\}$ on graph G_r (see, Section 2.1).

We estimate $\{\sigma(A\cup\{v\});\ v\in V\setminus A\}$ in the following way: First, we specify a sufficiently large positive integer M. Next, we independently generate a set $\{r_1,\cdots,r_M\}$ of M sample vectors on R_G from the probability distribution $q(r)$; that is, independently generate a set $\{G_{r_m};m=1,\cdots,M\}$ of M graphs. For any $v\in V\setminus A$, we approximate $\sigma(A\cup\{v\})$ by

$$\sigma(A\cup\{v\})\simeq\frac{1}{M}\sum_{m=1}^{M}|F(A\cup\{v\};G_{r_m})|. \tag{3}$$

Thus, we estimate $\{\sigma(A\cup\{v\});\ v\in V\setminus A\}$ on the basis of Equation (3) as follows:

1. **for** $m=1$ to M **do**
2. Generate graph G_{r_m}.
3. Compute $\{|F(A\cup\{v\};G_{r_m})|;\ v\in V\setminus A\}$.
4. Set $x_{v,m}\leftarrow|F\,(A\cup\{v\};G_{r_m})|$ for all $v\in V\setminus A$.
5. **end for**
6. Set $\sigma(A\cup\{v\})\leftarrow(1/M)\sum_{m=1}^{M}x_{v,m}$ for all $v\in V\setminus A$.

In particular, we evaluate $\{|F(A\cup\{v\};G_r)|;\ v\in V\setminus A\}$ for an arbitrary $r\in R_G$ by the following algorithm:

(E1) Find the subset $F(A;G_r)$ of V.

(E2) Set $|F(A\cup\{v\};G_r)|\leftarrow|F(A;G_r)|$ for all $v\in F(A;G_r)\setminus A$.

(E3) Find the subset $V_r^A=V\setminus F(A;G_r)$ of V, and the induced graph G_r^A of G_r to V_r^A.

(E4) Set $U\leftarrow\emptyset$.

(E5) while $V_r^A \setminus U \neq \emptyset$ **do**

(E6) Pick a node $u \in V_r^A \setminus U$.

(E7) Find the subset $F(u; G_r^A)$ of V_r^A.

(E8) Find the subset $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ of $F(u; G_r^A)$.

(E9) Set $|F(A \cup \{v\}; G_r)| \leftarrow |F(u; G_r^A)| + |F(A; G_r)|$ for all $v \in C(u; G_r^A)$.

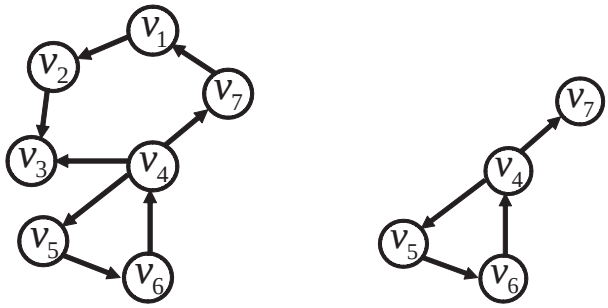
(E10) Set $U \leftarrow U \cup C(u; G_r^A)$.

(E11) end while

Now, we explain this algorithm. In Step (E1), we find the subset $F(A; G_r)$ that is reachable from A on graph G_r. In Step (E2), we use the fact that if $v \in F(A; G_r)$, the set $F(A \cup \{v\}; G_r)$ that is reachable from $A \cup \{v\}$ on G_r is equal to the set $F(A; G_r)$, and we simultaneously compute $|F(A \cup \{v\}; G_r)|$ for all $v \in F(A; G_r)$. In Step (E3), we find the subset $V_r^A = V \setminus F(A; G_r)$, and also find the induced graph G_r^A of graph G_r to V_r^A. In Steps (E4) to (E11), we use the fact that if $v \notin F(A; G_r)$, $|F(A \cup \{v\}; G_r)|$ is obtained by the sum of $|F(A; G_r)|$ and $|F(v; G_r^A)|$. This fact enables us to reduce the graph in question from G_r to G_r^A. We attempt to decompose graph G_r^A into its SCCs. In Step (E6), on graph G_r^A, we pick a node u that does not belong to the SCCs that we have already found. In Step (E7), we find the set $F(u; G_r^A)$ that is reachable from u on graph G_r^A. In Step (E8), we find the subset $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ of $F(u; G_r^A)$ by tracing backward all the links from u on the induced graph of G_r^A to $F(u; G_r^A)$. Note that the set $C(u; G_r^A)$ is equal to the SCC $SCC(u; G_r^A)$ that contains u. In Step (E9), we use the fact that $|F(v; G_r^A)| = |F(u; G_r^A)|$ if $v \in C(u; G_r^A)$, and simultaneously compute $|F(A \cup \{v\}; G_r)|$ for all $v \in C(u; G_r^A)$. We illustrate the flow of the algorithm in the following example:

Example: We consider the graph G_r shown in Figure 1a, where $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$. We set $A = \{v_1\}$. In this case, the process of the algorithm proceeds as follows.

In Step (E1), we find $F(A; G_r) = \{v_1, v_2, v_3\}$. In Step (E2), we find $|F(A \cup \{v_2\}; G_r)| = |F(A \cup \{v_3\}; G_r)| = 3$. In Step (E3), we find $V_r^A = \{v_4, v_5, v_6, v_7\}$ and G_r^A as shown in Figure 1b. In Step (E4), we set $U = \emptyset$. In Step (E5), we check $V_r^A \setminus U = \{v_4, v_5, v_6, v_7\} \neq \emptyset$. In Step (E6), we pick $v_4 \in V_r^A \setminus U$. In Step (E7), we find $F(v_4; G_r^A) = \{v_4, v_5, v_6, v_7\}$. In Step (E8), we find $C(v_4; G_r^A) = B(v_4; G_r^A) \cap F(v_4; G_r^A) = \{v_4, v_5, v_6\}$ in $F(v_4; G_r^A)$. In Step (E9), we find $|F(A \cup \{v_4\}; G_r)| = |F(A \cup \{v_5\}; G_r)| = |F(A \cup \{v_6\}; G_r)| = 7$. In Step (E10), we set $U = \{v_4, v_5, v_6\}$. In Step (E11), we return to Step (E5). In Step (E5), we check $V_r^A \setminus U = \{v_7\} \neq \emptyset$. In Step (E6), we pick $v_7 \in V_r^A \setminus U$. In Step (E7), we find $F(v_7; G_r^A) = \{v_7\}$. In Step (E8), we find $C(v_7; G_r^A) = \{v_7\}$. In Step (E9), we find $|F(A \cup \{v_7\}; G_r)|$



(a) An example of graph G_r.

(b) Graph G_r^A.

Figure 1: An illustration of the flow of the proposed algorithm for evaluating $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$, where $r \in R_G$ and $A = \{v_1\}$.

$= 4$. In Step (E10), we set $U = \{v_4, v_5, v_6, v_7\}$. In Step (E11), we return to Step (E5). In Step (E5), we check $V_r^A \setminus U = \emptyset$. Then, the process of the algorithm ends.

4.3 Computational Complexity of Proposed Method

In the same way as in Section 3.3, we evaluate the computational complexity of the proposed method as the expected number of examined nodes for estimating all the marginal influence degrees $\{\sigma(A \cup \{v\});\, v \in V \setminus A\}$ of A in Step (G3) of the greedy algorithm.

Let G_r be a graph generated from the occupation probability distribution $q(r)$ of the corresponding bond percolation model. We consider evaluating the expected number $\overline{Z(A, G_r)}$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ by the proposed method (see, Section 4.2). First, the number of examined nodes for finding $F(A; G_r)$ is given by $|F(A; G_r)|$. Let

$$V_r^A = \bigcup_{u \in U_r^A} SCC(u; G_r^A)$$

be the SCC decomposition of the induced graph G_r^A of G_r to $V_r^A = V \setminus F(A; G_r)$, where U_r^A stands for the set of all the representative nodes for SCCs. For any $u \in U_r^A$, the number of examined nodes for finding $F(u; G_r^A)$ is $|F(u; G_r^A)|$. Suppose now that $F(u; G_r^A)$ is found. Then, the number of examined nodes for finding $C(u; G_r^A)$ $(= SCC(u; G_r^A))$ is $|SCC(u; G_r^A)|$, since $C(u; G_r^A) = B(u; G_r^A) \cap F(u; G_r^A)$ is calculated on the induced graph of graph G_r^A to $F(u; G_r^A)$. Therefore, the number $Z(A, G_r)$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ by the proposed method is as follows:

$$Z(A, G_r) = |F(A; G_r)| + \sum_{u \in U_r^A} \left(|F(u; G_r^A)| + |SCC(u; G_r^A)| \right).$$

By the definition of graph G_r^A, we have

$$\sum_{u \in U_r^A} |SCC(u; G_r^A)| = N - |F(A; G_r)|,$$

where $N = |V|$. Thus, we have

$$Z(A, G_r) = N + \sum_{u \in U_r^A} |F(u; G_r^A)|. \tag{4}$$

Since $|F(u; G_r^A)| = |F(A \cup \{u\}; G_r)| - |F(A; G_r)|$, we can estimate the expected value of $|F(u; G_r^A)|$ as $\sigma(A \cup \{u\}) - \sigma(A)$. Hence, by Equation (4), we can estimate the expected number $\overline{Z(A, G_r)}$ of examined nodes for computing $\{|F(A \cup \{v\}; G_r)|;\, v \in V \setminus A\}$ as

$$\overline{Z(A, G_r)} = N + \left\langle \sum_{u \in U_r^A} (\sigma(A \cup \{u\}) - \sigma(A)) \right\rangle_r,$$

where $\langle f(r) \rangle_r$ stands for the operation that averages $f(r)$ with respect to r under $q(r)$, that is,

$$\langle f(r) \rangle_r = \sum_{r \in R(G)} f(r)\, q(r).$$

From the above results, we can estimate that the expected number $\mathcal{C}_1$ of examined nodes for the proposed method is

$$\mathcal{C}_1 = M \left\{ N + \left\langle \sum_{u \in U_r^A} (\sigma(A \cup \{u\}) - \sigma(A)) \right\rangle_r \right\}. \tag{5}$$

4.4 Computational Complexity Comparison

We compare the proposed method with the conventional method in terms of computational complexity. Both methods need M to be specified as a parameter, and we use the same value for both. We note that more coin-flips are used in the conventional method. In fact, if we think of a single run, i.e., any one of the M runs, the expected number of coin-flips for the conventional method is $O(|V|\sigma(v))$ for both the IC and LT models, whereas that for the proposed method is $O(|E|)$ for the IC model and $O(|V|)$ for the LT model. Note that in case of LT model for the proposed method, the coin-flip is realized by roulette for each node, i.e., picking at most one incoming link. However, if we focus on a single node v for initial activation from which to propagate the information, the number of coin-flips are $O(\sigma(v))$ for both the conventional and the proposed methods and for both the IC and the LT models because only the activated nodes (the expected number is $\sigma(v)$) are on the paths that lead to reachable nodes from v in the proposed

method. Thus by using the same value of M, both would estimate $\sigma(v)$ with the same accuracy in principle (see Appendix A). The biggest difference is that in the conventional method, when A is not empty, many of the coin-flips are redundant; that is, the diffusion process from A is repeatedly performed, whereas in the proposed method, no such repetition is made. This contributes to the stability of the proposed method. Below we begin by explaining the reason why we investigate the examined nodes to compare the proposed and the conventional methods.

First, we consider the case of IC model. Both the proposed and the conventional methods flip a coin with a bias $p_{u,v}$ on a link (u, v) to decide whether to propagate the information through the link (u, v) or not. Here, if we assume that all the coins are flipped in advance for the conventional method and ignore the computational complexity for flipping a coin and deciding whether or not to propagate the information, then for both the proposed and the conventional methods, the major computation is to trace forward or backward the links the information propagates and identify the nodes to visit. Therefore, we evaluate the computational complexities of the both methods for the IC model in terms of the expected number of examined nodes.

Next, we consider the case of LT model. For the proposed method, we ignore the computational complexity for the process of choosing at most one incoming link of each node in the original graph. For the conventional method, we ignore the computational complexity for the process of choosing the threshold θ_v of each node v in the original graph. Note that the proposed method performs the process M times, whereas the conventional method performs the process MN times. Moreover, for the conventional method, we further ignore the computational complexity for adding the weights from the neighboring active nodes to a node and deciding whether the node becomes active or not. Then, the major computation for the conventional method is to trace forward the links the information propagates and identify the nodes to visit. Therefore, we also evaluate the computational complexities of the both methods for the LT model in terms of the expected number of examined nodes.

Now, we compare the proposed and the conventional methods in terms of the expected number of examined nodes. We use the results in Sections 3.3 and 4.3. By Equation (2), the expected number $\mathcal{C}_0$ of examined nodes for the conventional method can be estimated as

$$\mathcal{C}_0 = M \left\{ N - |A| + \sum_{u \in V \setminus A} (\sigma(A \cup \{u\}) - 1) \right\}, \tag{6}$$

since $\sum_{V \setminus A} 1 = N - |A|$. In Equation (6), we can expect that $|A| \ll N$ $(= |V|)$, and $\sigma(A \cup \{u\}) - 1$ is summed up for almost all $u \in V$, since $k \ll N$. On the other hand, we can generally expect $|U_r^A| \ll N$ in Equation (5).

Also, we have $\sigma(A) > 1$ in the greedy algorithm if $A \neq \emptyset$. Moreover, for any $u \in V \setminus A$, $\sigma(A \cup \{u\}) - \sigma(A)$ decreases as $|A|$ increases, since $\sigma(A)$ is a submodular function. Hence, we can generally expect that in Step (G3) of the greedy algorithm, the proposed method has much smaller expected number of examined nodes than the conventional method.

From the above results, we can expect that compared with the conventional method, the proposed method will achieve a large reduction in computational cost.

5 Experimental Evaluation

Using large-scale real networks, we experimentally evaluated the performance of the proposed method.

5.1 Network Datasets

In the evaluation experiments, we should desirably use large-scale networks that exhibit many of the key features of real social networks. Here, we show the experimental results for two different datasets of such real networks.

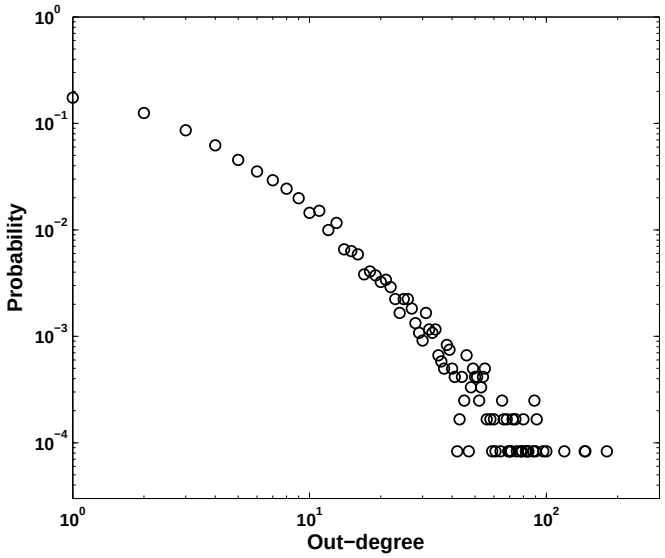


Figure 2: The out-degree distribution for the blog dataset.

First, we employed a trackback network of blogs, since a piece of information can propagate from one blog author to another blog author through a trackback, where a trackback is a kind of hyperlink with a *linkback* (i.e., link notification) function. We exploited the blog "Theme salon of blogs"

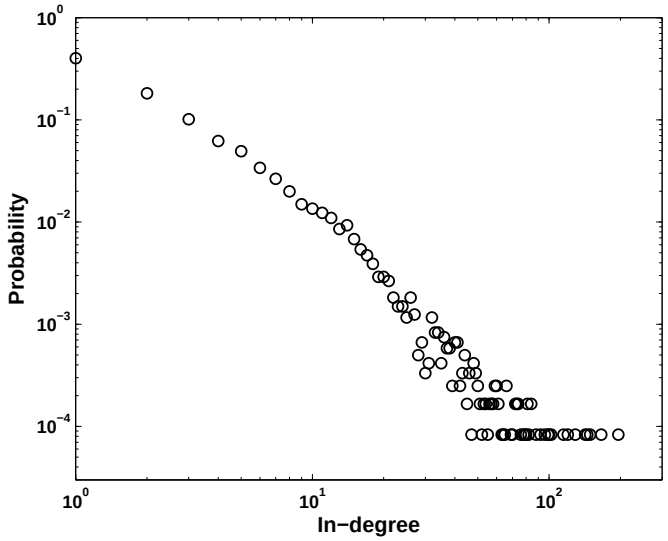


Figure 3: The in-degree distribution for the blog dataset.

in the site "goo" (`http://blog.goo.ne.jp/usertheme/`), where blog authors could recruit trackbacks of other blog authors by registering interesting themes. We collected a large-scale connected trackback network in May, 2005 by the following breadth first search process:

1. We started the process from the blog of the theme "JR Fukuchiyama Line Derailment Collision" in the site "goo", analyzed its HTML file, and extracted the list of the URLs of the source blogs of the trackbacks to this blog.

2. For each list obtained, we collected the blogs of the URLs in the list.

3. For each blog collected, we analyzed its HTML file, and constructed the list of the URLs of the source blogs of the trackbacks to the blog.

4. We repeated from Step 2 until depth ten from the original blog.

We call this network data the blog dataset. This network was a directed graph of $12,047$ nodes and $53,315$ links, and is expected to have a feature of real world social network in light of the way it is generated. To confirm this, the out-degree and in-degree distributions are respectively plotted in Figures 2 and 3, from which it is understood that these are "heavy-tailed" distributions that most large real networks exhibit. Here, the out-degree and in-degree distributions are the distributions of the number of outgoing and incoming links for every node, respectively. Thus, we believe that the blog dataset is a typical example of a large real social network represented

by a directed graph, and can be used as the network data to evaluate the performance of the proposed method.

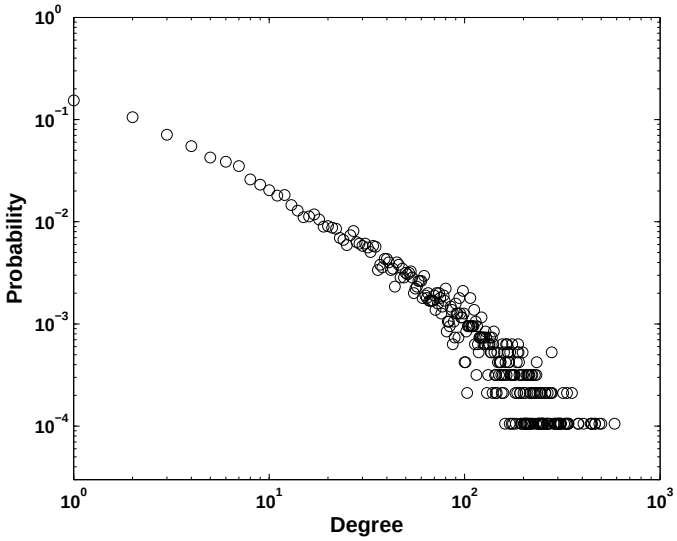


Figure 4: The degree distribution for the Wikipedia dataset.

Next, we employed a network of people that was derived from the "list of people" within Japanese Wikipedia. Specifically, we extracted the maximal connected component of the undirected graph obtained by linking two people in the "list of people" if they co-occur in six or more Wikipedia pages, and constructed a directed graph by regarding those undirected links as bidirectional ones. We call this network data the Wikipedia dataset. The total numbers of nodes and directed links were $9,481$ and $245,044$, respectively. Compared with the blog network, the way this network is generated is rather synthetically. Figure 4 shows the degree distribution of the undirected graph. We also observe that the degree distribution is a "heavy-tailed" distribution.

For social networks represented as undirected graphs, Newman and Park (2003) observed that they generally have the following two statistical properties that non-social networks do not have. First, they show positive correlations between the degrees of adjacent nodes. Second, they have much higher values of the *clustering coefficient* than the corresponding *configuration models* (i.e., random network models). Here, the clustering coefficient C for an undirected graph is defined by

$$C = \frac{3 \times \text{number of triangles on the graph}}{\text{number of connected triples of nodes}},$$

where a "triangle" means a set of three nodes each of which is connected to each other, and a "connected triple" means a node connected directly to

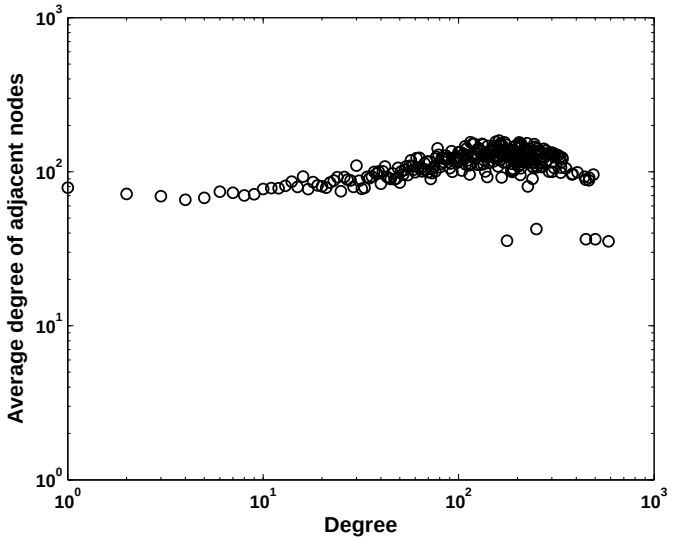


Figure 5: The degree correlation for the Wikipedia dataset.

unordered other pair nodes. Note that in terms of sociology, C measures the probability that two of your friends will also be friends each other. Given a degree distribution $\{\lambda_d\}$, the corresponding configuration model of a random network of N nodes is defined as the ensemble of all possible undirected graphs of N nodes that possess the degree distribution $\{\lambda_d\}$, where λ_d is the fraction of nodes in the network having degree d. It is known [18] that the value of C for the configuration model is exactly calculated by

$$C \;=\; \frac{1}{Nz_1}\left(\frac{z_2}{z_1}\right)^2,$$

where

$$z_1 \;=\; \sum_d \; d\lambda_d$$

is the average number of neighbors of a node and

$$z_2 \;=\; \sum_d \; d^2\lambda_d \;-\; \sum_d \; d\lambda_d$$

is the average number of second neighbors. For the undirected graph of the Wikipedia dataset, the value of C of the corresponding configuration model was 0.046, while the actual measured value of C was 0.39. Namely, the undirected graph of the Wikipedia dataset had a much higher value of the clustering coefficient than the corresponding configuration model. Moreover, we can see from Figure 5 that the Wikipedia dataset had weakly positive degree correlation. Therefore, we believe that the Wikipedia dataset is also

a typical example of a large real social network represented by an undirected graph, and can be used as the network data to evaluate the performance of the proposed method.

5.2 Experimental Settings

The proposed and the conventional methods are equipped with parameter M. We refer to the conventional method with $M = 1,000$ for the IC model as the *IC1000*. In the same way, we define the *LT1000* and *LT10000* for the conventional method with the LT model. We also refer to the proposed method using $M = 1,000$ and $M = 10,000$ for the IC model as the *ICBP1000* and *ICBP10000*, respectively. In the same way, we define the *LTBP1000* and *LTBP10000* for the proposed method with the LT model. As described in Section 4.4, we compare these methods for the same value of M.

The IC and LT models have parameters to be specified in advance. In the IC model, we assigned a uniform probability p to the propagation probability $p_{u,v}$ for any directed link (u, v) of the network, that is, $p_{u,v} = p$. In the LT model, we uniformly set weights as follows: For any node v of the network, the weight $w_{u,v}$ from a parent node $u \in \Gamma(v)$ is given by $w_{u,v} = 1/|\Gamma(v)|$.

We implemented all our programs of both the conventional and proposed methods for the IC and LT models in C language. Of course, the basic structure of these programs is the same, except that the routines of active node calculation used in the conventional method are replaced with those of bond percolation and SCC decomposition used in the proposed method.

5.3 Experimental Results

We compared the proposed method with the conventional method in terms of both the performance of the approximate solution A_k and the processing time for solving the influence maximization problem of size k. The performance of A_k is measured by the influence degree $\sigma(A_k)$. We estimated $\sigma(A_k)$ by using $300,000$ simulations according to the work of Kempe et al. (2003). All our experimentation was undertaken on a single Dell PC with an Intel 3.4GHz Xeon processor, with 2GB of memory, running under Linux.

In order to keep computational time at a reasonable level for the conventional method, we mainly compared these two methods using $M = 1,000$. Note that if a large enough M is taken, these two methods should produce the same solution. We conjecture that $M = 1,000$ is not large enough, that is, these two methods with $M = 1,000$ cannot necessarily obtain good approximate values for the marginal influence degrees $\{\sigma(A \cup \{v\}); v \in V \setminus A\}$ of A, (see Appendices A and B). Thus, we iterated the same experiment five times independently. Tables 1 and 2 show the experimental results for the IC model with $p = 10\%$ and the LT model for the blog dataset, respectively, where the values are rounded to three significant figures. Note that

Table 1: Performance of approximate solutions for the influence maximization problem under the IC model with $p = 10\%$ for the blog dataset. Upper: IC1000 (the conventional method). Lower: ICBP1000 (the proposed method).

k	$\sigma(A_k)$ (IC1000)				
1	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2
10	6.93×10^2	6.98×10^2	6.93×10^2	6.91×10^2	6.95×10^2
20	8.58×10^2	8.61×10^2	8.57×10^2	8.58×10^2	8.60×10^2
30	9.59×10^2	9.69×10^2	9.68×10^2	9.66×10^2	9.78×10^2

k	$\sigma(A_k)$ (ICBP1000)				
1	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2	1.74×10^2
10	7.02×10^2	7.01×10^2	7.00×10^2	7.01×10^2	7.02×10^2
20	8.74×10^2	8.75×10^2	8.73×10^2	8.74×10^2	8.73×10^2
30	9.91×10^2	9.92×10^2	9.90×10^2	9.92×10^2	9.92×10^2

in these tables and later ones, too, the values are reestimated with $300,000$ simulations once A_k has been obtained by each method with a specified M. Since the true solution $\sigma(A_k^*)$ is by definition the maximum among all $\sigma(A_k)$, if $\sigma(A_k)$ is estimated accurately, it makes sense to argue that the larger the value is, the closer it is to the true solution and thus it is of better quality. We first observe that the results for the proposed method were relatively stable over the iterations, while the results for the conventional method somewhat fluctuated for large k in particular. Here, we note that the proposed method using $M = 10,000$ was stable and always produced the same solution for $k = 30$ over the iterations (not shown in the tables). We also observe that for $k = 30$, the solutions by the ICBP1000 and LTBP1000 outperforms those by the IC1000 and LT1000, respectively.

Table 3 shows the processing time to obtain A_k by the IC1000, ICBP1000, LT1000 and LTBP1000 for the blog dataset, where the values are rounded to three significant figures. We observe from Table 3 that the ICBP1000 and LTBP1000 are much more efficient than the IC1000 and LT1000, respectively. For example, to obtain the approximate solution A_{30} for $k = 30$, both the IC1000 and LT1000 needed about 2.5 days, while the ICBP1000 and LTBP1000 needed about 2.5 and 1.5 minutes, respectively. Namely, for $k = 30$, the ICBP1000 was 1.8×10^3 times faster than the IC1000, and the LTBP1000 was 4.6×10^3 times faster than the LT1000. We also examined the LT10000 and LTBP10000 on the blog dataset. In order to obtain approximate solution A_{30}, the LT10000 needed about 27 days, while the LTBP10000 needed only about 14 minutes.

Table 2: Performance of approximate solutions for the influence maximization problem under the LT model for the blog dataset. Upper: LT1000 (the conventional method). Lower: LTBP1000 (the proposed method).

k	$\sigma(A_k)$ (LT1000)				
1	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2
10	1.59×10^3	1.61×10^3	1.61×10^3	1.59×10^3	1.58×10^3
20	2.41×10^3	2.40×10^3	2.42×10^3	2.42×10^3	2.38×10^3
30	3.02×10^3	3.05×10^3	3.01×10^3	3.01×10^3	3.00×10^3

k	$\sigma(A_k)$ (LTBP1000)				
1	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2	2.86×10^2
10	1.60×10^3	1.61×10^3	1.61×10^3	1.59×10^3	1.60×10^3
20	2.44×10^3	2.44×10^3	2.44×10^3	2.44×10^3	2.44×10^3
30	3.07×10^3	3.07×10^3	3.06×10^3	3.06×10^3	3.06×10^3

Table 3: Processing time (sec.) for the blog dataset.

k	IC1000	ICBP1000	LT1000	LTBP1000
1	3.70×10^2	7.07	6.57×10^2	3.19
10	4.69×10^4	5.68×10^1	4.24×10^4	2.96×10^1
20	1.24×10^5	1.09×10^2	1.25×10^5	5.64×10^1
30	2.13×10^5	1.60×10^2	2.32×10^5	8.20×10^1

Tables 4, 5 and 6 show the experimental results for the Wikipedia dataset. We see that the results were qualitatively very similar to the ones for the blog dataset. First, the solutions by the ICBP1000 and LTBP1000 outperformed those by the IC1000 and LT1000, respectively. We also note that the proposed method using $M = 10,000$ was stable and always produced the same solution for $k = 30$ over the iterations (not shown in the tables). Next, the ICBP1000 and LTBP1000 were much more efficient than the IC1000 and LT1000, respectively. For example, for obtaining the approximate solution A_{30} for $k = 30$, the ICBP1000 was 1.9×10^3 times faster than the IC1000, and the LTBP1000 was 8.3×10^3 times faster than the LT1000. We also conducted experiments on some other large-scale real networks including a blogroll network of blogs, and confirmed the effectiveness of the proposed method.

Table 4: Performance of approximate solutions for the influence maximization problem under the IC model with $p = 1\%$ for the Wikipedia dataset. Upper: IC1000 (the conventional method). Lower: ICBP1000 (the proposed method).

k	$\sigma(A_k)$ (IC1000)				
1	1.39×10^2	1.39×10^2	1.36×10^2	1.36×10^2	1.36×10^2
10	3.91×10^2	3.97×10^2	3.98×10^2	4.02×10^2	4.01×10^2
20	4.56×10^2	4.64×10^2	4.62×10^2	4.64×10^2	4.66×10^2
30	4.97×10^2	5.02×10^2	4.95×10^2	5.00×10^2	4.98×10^2

k	$\sigma(A_k)$ (ICBP1000)				
1	1.39×10^2	1.39×10^2	1.39×10^2	1.36×10^2	1.36×10^2
10	4.05×10^2	4.06×10^2	4.07×10^2	4.06×10^2	4.07×10^2
20	4.75×10^2	4.76×10^2	4.76×10^2	4.75×10^2	4.77×10^2
30	5.16×10^2	5.17×10^2	5.17×10^2	5.16×10^2	5.17×10^2

5.4 Discussion

These experimental results show that the proposed method is much more efficient than the conventional method.

First, we investigate the reason why the proposed method outperforms the conventional method in the case of $M = 1,000$ for our network datasets. If we take a sufficiently large M (e.g., $M = 100,000$), the proposed and the conventional methods should produce the same solution. As shown in the experiments, the estimation accuracy of influence degree function σ with $M = 1,000$ is not so high for the both methods. Now, consider estimating all the marginal influence degrees $\{\sigma(A_k \cup \{v\});\ v \in V \setminus A_k\}$ of solution A_k, and choosing the node v_{k+1} that maximizes $\sigma(A_k \cup \{v\})$, $(v \in V \setminus A_k)$. It should be reemphasized that the influence set of A_k is equally evaluated for all $v \in V \setminus A_k$ for the proposed method. In fact, when $\sigma(A_k \cup \{v\})$ is estimated using Equation (3), each $|F(A_k \cup \{v\}; G_{r_m})|$ is basically computed by

$$|F(A_k \cup \{v\}; G_{r_m})| = \left|F(v; G_{r_m}^{A_k})\right| + |F(A_k; G_{r_m})|\,.$$

Thus, for the proposed method, a node that is relatively optimal for A_k can be selected as v_{k+1}. On the other hand, for the conventional method, the influence set of A_k is not equally evaluated for all $v \in V \setminus A_k$ since $\sigma(A_k \cup \{v\})$ is independently estimated for every v each by a distinct simulation. We also note that the number of final active nodes for a given target set greatly varied for every simulation in the IC and LT models (see, Appendix B). Thus, unlike the proposed method, the selection of v_{k+1} in the conventional method

Table 5: Performance of approximate solutions for the influence maximization problem under the LT model for the Wikipedia dataset. Upper: LT1000 (the conventional method). Lower: LTBP1000 (the proposed method).

k	$\sigma(A_k)$ (LT1000)				
1	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2
10	1.72×10^3	1.72×10^3	1.67×10^3	1.66×10^3	1.72×10^3
20	2.55×10^3	2.55×10^3	2.45×10^3	2.53×10^3	2.55×10^3
30	3.12×10^3	3.03×10^3	2.99×10^3	3.01×10^3	3.11×10^3

k	$\sigma(A_k)$ (LTBP1000)				
1	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2	3.41×10^2
10	1.72×10^3	1.72×10^3	1.72×10^3	1.72×10^3	1.71×10^3
20	2.58×10^3	2.58×10^3	2.59×10^3	2.59×10^3	2.59×10^3
30	3.18×10^3	3.18×10^3	3.18×10^3	3.18×10^3	3.18×10^3

Table 6: Processing time (sec.) for the Wikipedia dataset.

k	IC1000	ICBP1000	LT1000	LTBP1000
1	6.63×10^2	1.91×10^1	5.41×10^2	5.17
10	1.94×10^5	1.74×10^2	9.60×10^4	4.64×10^1
20	4.82×10^5	3.42×10^2	3.03×10^5	8.57×10^1
30	8.03×10^5	5.10×10^2	5.69×10^5	1.21×10^2

using $M = 1,000$ by necessity completely depends on how the influence set of A_k is evaluated by chance for each $v \in V \setminus A_k$. Therefore, we believe that the proposed method outperforms the conventional method in the case of $M = 1,000$ for our network datasets.

Here, to explain the point of the reason described above more clearly, we consider the following method as an extended version of the conventional method:

1. **for** $m = 1$ to M **do**
2. Find the set $D(A_k)$ of active nodes at the end of the random process of the IC or the LT models for initial active set A_k by simulation.
3. **for** each $v \in V \setminus A_k$ **do**
4. Find the set $D(v)$ of active nodes at the end of the random process of the IC or the LT models for initial active set $\{v\}$ by simulation.

5. Set $x_{v,m} \leftarrow |D(A_k) \cup D(v)|$.

6. **end for**

7. **end for**

8. **for** each $v \in V \setminus A_k$ **do**

9. Set $\sigma(A_k \cup \{v\}) \leftarrow (1/M) \sum_{m=1}^{M} x_{v,m}$

10. **end for**

The extended method should improve the conventional method because the influence set of A_k is now equally evaluated for all $v \in V \setminus A_k$, and should be comparable to the proposed method in quality of solution. However, it cannot be as efficient as the proposed method since it does not incorporate the SCC-finding technique.

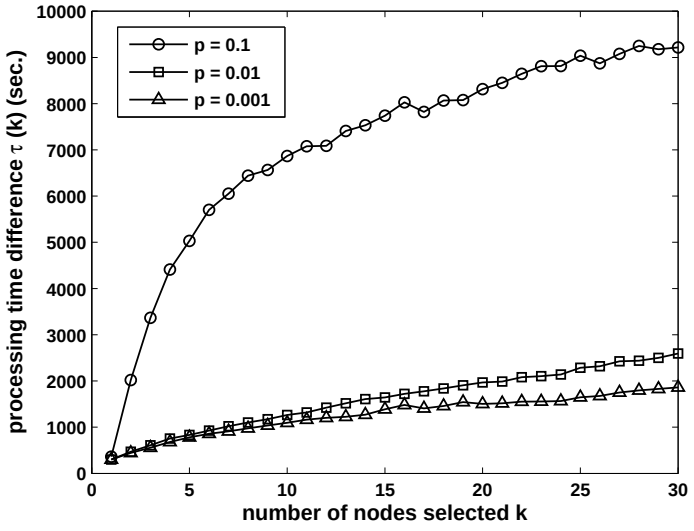


Figure 6: Processing time difference $\tau(k)$ between the proposed and conventional methods for the blog dataset in the case of the IC model.

Next, we discuss the sources of the difference between the proposed and conventional methods in processing time. Note that we use the same value of parameter M for both methods. Let $\tau_1(k)$ and $\tau_0(k)$ respectively denote the processing times of the proposed and the conventional methods for obtaining solution A_{k+1} when solution A_k is given. We define the processing time difference $\tau(k)$ by $\tau_0(k) - \tau_1(k)$ for k, the number of nodes selected. We believe the essential sources of speed-up in the proposed method is that we compute $\{|F(A_k \cup \{v\}; G_r)|;\ v \in V \setminus A_k\}$ on graph G_r as follows:

- By first identifying $F(A_k; G_r)$, we reduce the graph in question from G_r to the induced graph $G_r^{A_k}$ of G_r to $V \setminus F(A_k; G_r)$
- By decomposing $G_r^{A_k}$ into the SCCs, we compute $|F(A_k \cup \{v\}; G_r)|$ for many nodes v at once.

Namely, we believe that the larger the size of $F(A_k; G_r)$ is, the larger the value of $\tau(k)$ is. Moreover, we believe that the larger the sizes of the SCCs of graph $G_r^{A_k}$ are, the larger the value of $\tau(k)$ is. Here, we demonstrate these characteristics for the IC model. Note that the size of $F(A_k; G_r)$ monotonically increases with the value of k. Thus, we can expect that the value of $\tau(k)$ also monotonically increases with the value of k. Note also that graph G_r becomes denser when the value of the propagation probability p is larger, and the sizes of the SCCs of G_r also become larger. Thus, we can also expect that the value of $\tau(k)$ monotonically increases with the value of p. Figure 6 shows $\tau(k)$ for $p = 0.1\%$, 1% and 10% as a function of k for the blog dataset, where circles, squares and diamonds indicate $\tau(k)$ for $p = 0.1\%$, 1% and 10%, respectively. Here, we used $M = 1,000$ for both the proposed and the conventional methods. The results support our conjectures.

6 Related Work

6.1 Calculation of Influence Degrees

First, we describe work related to the calculation of influence degrees in the IC model. Let us recall that the SIR model for the spread of a disease on a network is equivalent to a bond percolation model on the same network, and the size of a disease outbreak from a node corresponds to the size of the cluster that can be reached from the node by traversing only the "occupied" links. There are a series of work that uses this correspondence to develop a method for theoretically calculating the probability distribution of the size of a disease outbreak that starts with a randomly chosen node in the configuration model (i.e., a random network model) with a given degree distribution (Callaway et al., 2000; Newman, 2002; Newman, 2003), and to derive a condition for the disease outbreak from a randomly chosen node to give an *epidemic outbreak* that affects a non-zero fraction on the network in the limit of very large network. Mathematically more rigorous treatments of similar results can be found in the work of Molloy and Reed (1998) and Chung and Lu (2002).

Next, we describe work related to the calculation of influence degrees in the LT model. Watts (2002) investigated the LT model on a network to explain large but rare cascade phenomena triggered by small initial shocks. Using the concept of *site percolation*, he theoretically derived a condition for the cascade from a randomly chosen seed node to give a *global cascade* that affects a non-zero fraction on the network in the limit of infinitely large

network for the configuration model (i.e., a random network model) with a given degree distribution.

The above mentioned studies focused on global properties averaged over a random network in the limit of very large size, while our primary interest is to practically answer which nodes are most influential for information diffusion on a given real-world network of a finite size. We also note that those studies dealt with undirected graphs, while our work investigates information diffusion on networks represented by directed graphs. Moreover, the theories developed in those studies assumed that the loop structure on a network of interest can be essentially ignored in the limit of large network size. However, this property is not true of many large-scale social networks, and it is an open question whether or not those theories are effective for such networks (Newman, 2003). In fact, the clustering coefficient C quantifies the loop structure in a network, and it was indeed observed that many social networks have much higher values of C than the corresponding configuration models (i.e., random network models) (Newman and Park, 2003).

6.2 Solving the Influence Maximization Problem

The influence degree function σ is submodular (see, Kempe et al., 2003). For solving a combinatorial optimization problem of a submodular function f on V by the greedy algorithm, Leskovec et al. (2007) have recently presented a lazy evaluation method that leads to far fewer (expensive) evaluations of the marginal increments $f(A \cup \{v\}) - f(A)$ $(v \in V \setminus A)$ in the greedy algorithm for $A \neq \emptyset$, and achieved an improvement in speed. Note here that their method requires evaluating $f(v)$ for all $v \in V$ at least. Thus, we can apply their method to the influence maximization problem for the IC or LT models, where the influence degree function σ is evaluated through the simulations of the corresponding random process. It is clear that this method is more efficient than the conventional method. However, the proposed method for $k = 30$ was faster than the conventional method for $k = 1$ as shown in Tables 3 and 6. Therefore, it is evident that the proposed method can be faster than the method by Leskovec et al. (2007) for the influence maximization problem for the IC or LT models. To quantify the difference we implemented the lazy evaluation method. The processing time for $k = 30$ in case of the blog dataset was 2.12×10^3 and 8.28×10^2 seconds for the IC and the LT models, respectively, and the corresponding processing time in case of Wikipedia dataset was 1.46×10^4 and 2.65×10^3 seconds for the IC and the LT models, respectively. Here, $M = 1,000$ are used as the number of simulations (see, Section 3.2), and the values are rounded to three significant figures. From these results, we can see that the proposed method was more than ten times faster than the method by Leskovec et al. (2007) for $k = 30$ in the blog and Wikipedia datasets (see, Tables 3 and 6).

Beyond the IC and LT models, Kempe et al. (2003) proposed the *trig-*

gering model as an yet another diffusion model on a network. It is proved that the triggering model can be identified with a bond percolation model (see, Kempe et al., 2003). The proposed method can be applied to this model because it can be applied to any diffusion model that can be identified with a bond percolation model. The future work includes presenting a large number of realistic examples of such diffusion models.

In this paper, we have considered the *progressive* case in which nodes cannot switch from being active to being inactive. However, there are many information diffusion phenomena that non-progressive diffusion models are required. Examples include the spread of posts for a topic in blogspace (Gruhl et al, 2004). Kempe et al. (2003) proved that *non-progressive* case can be reduced to the progressive case. More specifically, it is proved that the influence maximization problem for a non-progressive diffusion model on graph G in time-limit T is equivalent to the ordinary influence maximization problem on the *layered graph* G_T for the progressive diffusion model, where G_T is the directed acyclic graph (DAG) constructed by time-forwardly connecting $(T+1)$ copies of G (see, Kempe et al. 2003). Therefore, building effective methods for fundamental progressive models such as the IC and LT models is indeed important and crucial for the non-progressive case.

From a realistic point of view, the IC and LT models are by no means a complete model, but are at best a simplified and partial representation of a complex reality (see, Kempe et al, 2003; Gruhl et al., 2004; Leskovec et al., 2006). However, in the field of sociology, Watts and Dodds (2007) recently examined the "influentials hypothesis" in the contexts of the LT model and the SIR model (i.e., an extended model of the IC model), that is, they investigated by computer simulations whether large cascades of influence are actually driven by influentials or not. On the other hand, Even-Dar and Shapira (2007) mathematically studied the influence maximization problem in the context of another fundamental model called the voter model. We also believe that it is important to investigate information diffusion phenomena for the IC and LT models (i.e., fundamental diffusion models) to deepen our understanding of these models. The future work includes proposing effective methods for solving the influence maximization problem in the contexts of various realistic diffusion models.

6.3 Applications

As is easily understood, the conventional method is not practical unless we rely on high-performance computers and sophisticated techniques such as parallel computing (see, Tables 3 and 6) to solve the kind of problems such as influence maximization problem as addressed in this paper. In contrast, the proposed method enables us to obtain a practical solution to this kind of problems on a single standard PC in a reasonable processing time. Thus, we can apply the proposed method to a variety of real problems.

The work of Watts and Dodds (2007) briefly described above needs a method to efficiently estimate $\sigma(A)$ and the proposed method can readily be applicable.

As mentioned in the introduction, the influence maximization problem finds many realistic applications. The most straightforward application would be viral marketing. When we wish to promote a new product (e.g., an email service or a search engine), and are given a relevant social network, we can easily find a limited number of key (influential) persons first to adopt the new product by the proposed method, and enjoy the diffusion effect for the IC or LT models (i.e., fundamental diffusion models) through the social network. We admit that the diffusion models we discussed are oversimplified but still it is useful to obtain approximate solutions as a first step toward an effective marketing without using classical advertising channels.

The proposed method has an application of different flavor which is the visualization of information flow. Understanding the flow of information through a complex network is important in terms of sociology and marketing. We devised a new node embedding method for visualizing the information diffusion process from the target nodes selected to be a solution of the influence maximization problem (Saito et al., 2008). This visualization method is characterized by 1) utilization of the target nodes as a set of pivot objects for visualization, 2) application of a probabilistic algorithm for embedding all the nodes in the network into an Euclidean space to conserve the posterior information diffusion probability, and 3) varying appearance of the embedded nodes on the basis of two label assignment strategies, one with emphasis on influence of initially activated nodes, and the other on degree of information reachability.

7 Conclusion

We have considered the influence maximization problem for the IC and LT models on a large-scale social network represented as a directed graph $G = (V, E)$. Due to the computational complexity, the greedy search algorithm is the only practical approach, but still the conventional method needed a high amount of computation. We have proposed a method of efficiently finding a good approximate solution to the problem under the greedy algorithm. In particular, in order to improve the computational efficiency, we have estimated all the marginal influence degrees $\{\sigma(A \cup \{v\});\ v \in V \setminus A\}$ of a given target set A in the following way:

- We identify the IC and LT models with the corresponding bond percolation models.
- For any $v \in V \setminus A$, we estimate the influence degree $\sigma(A \cup \{v\})$ of $A \cup \{v\}$ as the empirical mean of the number $|F(A \cup \{v\}; G_r)|$ of the

nodes that are reachable from $A \cup \{v\}$ on a graph G_r generated from the corresponding occupation probability distribution $q(r)$ of the bond percolation.

In particular, we estimate $\{|F(A \cup \{v\}; G_r)|;\ v \in V \setminus A\}$ as follows:

- We find the set $F(A; G_r)$ that is reachable from A on graph G_r, and simultaneously compute $\{|F(A \cup \{v\}; G_r)|;\ v \in F(A; G_r)\}$.
- We find the induced graph G_r^A of G_r to $V \setminus F(A; G_r)$, and decompose G_r^A into its SCCs (Strongly Connected Components).
- For each SCC $SCC(u; G_r^A)$ of G_r^A, $(u \in V \setminus F(A; G_r))$, we simultaneously compute $\{|F(A \cup \{v\}; G_r)|;\ v \in SCC(u; G_r^A)\}$.

We have compared the proposed method with the conventional method in terms of computational complexity and quality of the solution, and have shown that the proposed method is expected to achieve a large amount of reduction in computational cost. Moreover, using large-scale networks including a real blog network, we have experimentally demonstrated the effectiveness of the proposed method. For example, we obtained the following results for the influence maximization problem of size $k = 30$ on the blog and Wikipedia datasets that are real networks with about $10,000$ nodes: In the case of the IC model, the proposed method was 1800 times faster than the conventional method, and in the case of the LT model, the proposed method was 4600 times faster than the conventional method.

Acknowledgement

This work was partly supported by JSPS Grant-in-Aid for Scientific Research (C) (No. 20500147), and Asian Office of Aerospace Research and Development, Air Force Office of Scientific Research, U.S. Air Force Research Laboratory under Grant No. AOARD-08-4027.

Appendix

A Convergence Speed

As described in Section 4.4, by using the same value of M, both the proposed and the conventional methods would estimate $\sigma(v)$ with the same accuracy in principle. Here, we experimentally demonstrate this conjecture.

According to the work of Kempe et al. (2003), we set $M = 300,000$ as a sufficiently large value of M, that is, we assume that $\sigma(v)$ for any $v \in V$ is well approximated by $300,000$ simulations of the information diffusion model (i.e., the conventional method using $M = 300,000$). For any $v \in V$, let $\sigma_0(v; M)$ and $\sigma_1(v; M)$ denote the estimates of $\sigma(v)$ by the conventional and the proposed methods using parameter value M, respectively. For the blog and Wikipedia datasets, we investigated

$$\mathcal{E} = \frac{1}{N} \sum_{v \in V} |\sigma_0(v; 300, 0000) - \sigma_1(v; 300, 000)|,$$

$$\mathcal{E}_0(M) = \frac{1}{N} \sum_{v \in V} |\sigma_0(v; M) - \sigma_0(v; 300, 000)|,$$

$$\mathcal{E}_1(M) = \frac{1}{N} \sum_{v \in V} |\sigma_1(v; M) - \sigma_1(v; 300, 000)|.$$

We first consider the case of the IC model. Then, the value of $\mathcal{E}$ was 0.03 and 0.04 for the blog and Wikipedia datasets, respectively. Thus, we can assume that the values of $\sigma_0(v; 300,000)$ and $\sigma_1(v; 300,000)$ are almost the same for any $v \in V$.

Table 7: Convergence speed for the blog dataset.

M	$\mathcal{E}_0(M)$	$\mathcal{E}_1(M)$
100	1.16	1.12
1,000	0.36	0.36
10,000	0.11	0.12
100,000	0.03	0.03

Table 8: Convergence speed for the Wikipedia dataset.

M	$\mathcal{E}_0(M)$	$\mathcal{E}_1(M)$
100	1.28	1.23
1,000	0.42	0.42
10,000	0.13	0.14
100,000	0.03	0.03

Tables 7 and 8 show the values of $\mathcal{E}_0(M)$ and $\mathcal{E}_1(M)$ for the blog and Wikipedia datasets, respectively. These results imply that the proposed and the conventional methods estimate $\{\sigma(v);\, v \in V\}$ with almost the same

accuracy for the IC model. We also obtained similar results for the case of the LT model. For example, the value of $\mathcal{E}$ was 0.03 and 0.09 for the blog and Wikipedia datasets, respectively. For the blog dataset, the values of $\mathcal{E}_0(10,000)$ and $\mathcal{E}_1(10,000)$ were 0.13 and 0.12, respectively. Also, for the Wikipedia datasets, the values of $\mathcal{E}_0(10,000)$ and $\mathcal{E}_1(10,000)$ were 0.36 and 0.37, respectively. These results support our conjecture.

B Fluctuation in Simulations of Information Diffusion Models

For each $v \in V$, we examine fluctuation in the number $\varphi(v)$ of the final active nodes for a target initially activated node v through $1,000$ simulations in the IC and LT models. Let $\mu(v)$ and $s(v)$ denote the empirical mean and the standard deviation of $\varphi(v)$ for $1,000$ simulations, respectively. We define $\overline{\mu}$ and $\overline{s}$ by the empirical means of $\{\mu(v);\ v \in V\}$ and $\{s(v);\ v \in V\}$, respectively. For the blog dataset, $\overline{\mu}$ and $\overline{s}$ were as follows:

IC model ($p = 10\%$): $\overline{\mu} = 8.6$, $\overline{s} = 14.3$.

LT model: $\overline{\mu} = 6.8$, $\overline{s} = 14.9$.

For the Wikipedia dataset, $\overline{\mu}$ and $\overline{s}$ were as follows:

IC model ($p = 1\%$): $\overline{\mu} = 8.1$, $\overline{s} = 16.1$,

LT model: $\overline{\mu} = 12.6$, $\overline{s} = 42.4$,

Here, the values are rounded to the first decimal place. We can observe that compared with $\overline{\mu}$, $\overline{s}$ is very large. Therefore, we see that the number of final active nodes for a given target set can greatly vary for every simulation in the IC and LT models.

References

[1] Callaway, D. S., Newman, M. E. J., and Strogatz, S. H. 2000. Network robustness and fragility: Percolation on random graphs. Physical Review Letters, 85:5468–5471.

[2] Chung, F. and Lu, L. 2002. Connected components in a random graph with given expected degree sequences. Annals of Combinatorics, 6:125-145.

[3] Domingos, P. 2005. Mining social networks for viral marketing. IEEE Intelligent Systems, 20:80–82.

[4] Domingos, P. and Richardson, M. 2001. Mining the network value of customers. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, pp. 57–66.

[5] Even-Dar, E. and Shapira, A. 2007. A note on maximizing the spread of influence in social networks. Internet and Network Economics: WINE 2007, LNCS 4858, pp. 281–286.

[6] Goldenberg, J., Libai, B., and Muller, E. 2001. Talk of the network: A complex systems look at the underlying process of word-of-mouth. Marketing Letters, 12:211–223.

[7] Grassberger, P. 1983. On the critical behavior of the general epidemic process and dynamical percolation. Mathematical Bioscience, 63:157–172.

[8] Gruhl, D., Guha, R., Liben-Nowell, D., and Tomkins, A. 2004. Information diffusion through blogspace. Proceedings of the 7th International World Wide Web Conference, New York, USA, pp. 107–117.

[9] Kempe, D., Kleinberg, J., and Tardos, E. 2003. Maximizing the spread of influence through a social network. Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, pp. 137–146.

[10] Kempe, D., Kleinberg, J., and Tardos, E. 2005. Influential nodes in a diffusion model for social networks. Automata, Languages and Programming: ICALP 2005, LNCS 3580, pp. 1127–1138.

[11] Leskovec, J., Adamic, L. A., and Huberman, B. A. 2006. The dynamics of viral marketing. Proceedings of the 7th ACM Conference on Electronic Commerce, Ann Arbor, Michigan, USA, pp. 228–237.

[12] Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., and Glance, N. 2007. Cost-effective outbreak detection in networks. Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, pp. 420–429.

[13] McCallum, A., Corrada-Emmanuel, A., and Wang, X. 2005. Topic and role discovery in social networks. Proceedings of the 19th International Joint Conference on Artificial Intelligence, Edinburugh, Scotland, pp. 786–791.

[14] Molloy, M. and Reed, B. 1998. The size of the giant component of a random graph with a given degree sequence. Combinatorics, Probability and Computing, 7:295–305.

[15] Newman, M. E. J. and Park, J. 2003. Why social networks are different from other types of networks. Physical Review E, 68:036122.

[16] Newman, M. E. J. 2001. The structure of scientific collaboration networks. Proceedings of the National Academy of Sciences of the United States of America, 98:404–409.

[17] Newman, M. E. J. 2002. Spread of epidemic disease on networks. Physical Review E, 66:016128.

[18] Newman, M. E. J. 2003. The structure and function of complex networks. SIAM Review, 45:167–256.

[19] Richardson, M. and Domingos, P. 2002. Mining knowledge-sharing sites for viral marketing. Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton, Alberta, Canada, pp. 61–70.

[20] Saito, K., Kimura, M., and Motoda, H. 2008. Effective visualization of information diffusion process over complex networks. Machine Learning and Knowledge Discovery in Databases: ECML PKDD 2008, LNAI 5212, pp. 326–341.

[21] Watts, D. J. 2002. A simple model of global cascades on random networks. Proceedings of the National Academy of Sciences of the United States of America, 99:5766–5771.

[22] Watts, D. J. and Dodds, P. S. 2007. Influence, networks, and public opinion formation. Journal of Consumer Research, 34:441–458.